Spring 2009

Towards the Quantitative Analysis of the Connectivity Value of Networked Operations

Dean Deller
Old Dominion University

TOWARDS THE QUANTITATIVE ANALYSIS OF THE CONNECTIVITY VALUE OF NETWORKED OPERATIONS

by

Sean Deller
B.S. May 1988, United States Military Academy
M.E. December 1999, Old Dominion University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

DOCTOR OF PHILOSOPHY

ENGINEERING MANAGEMENT

OLD DOMINION UNIVERSITY
May 2009

Approved by:

Ghaith Rabadi (Director)

Shannon Bowling (Member)

Andreas Tolk (Member)

Steven Mains (Member)

ABSTRACT

TOWARDS THE QUANTITATIVE ANALYSIS OF THE CONNECTIVITY VALUE OF NETWORKED OPERATIONS

Sean Deller
Old Dominion University, 2009
Director: Dr. Ghaith Rabadi

While the nature of and the approach to command and control is evolving in order to meet the challenges of Information Age warfare, the essential functions that must be accomplished remain constant. One of those essential functions is the arrangement of the assets within a combat force. Certainly, the many different ways to arrange a given set of assets will have different impacts on the combat effectiveness of the force. Some arrangements will enable self-synchronization, while other arrangements will impede it. How then, should an Information Age combat force be organized in order to optimize its effectiveness?

The concept of Network Centric Operations (NCO) represents a shift from traditional attrition-based approaches to a warfighting style that emphasizes speed of command and self-synchronization. One goal of NCO is to field a force that is capable of achieving information superiority, thus enabling a massing of effects instead of the traditional massing of forces that will disrupt the enemy's strategy and preclude potential courses of action. NCO shifts the focus from the numbers and capabilities of platforms toward the information-based aspects of force employment: information collection, communication, and exploitation. Central to the ability of a force to manage and exploit information is its connectivity: the existence, capacity, reliability, and other attributes of the links that connect its platforms, command and control centers, and other entities. A fundamental problem is to develop an understanding of the influence of connectivity on force effectiveness that can lead eventually to quantitative prediction and guidelines for design and employment. This research presents an initial attempt to achieve such understanding through the quantitative analysis of a model of NCO focused on the correlation between connectivity and effectiveness.

To Pam, Brendan, and Thomas...

ACKNOWLEDGEMENTS

I am deeply grateful to Professors Ghaith Rabadi, Shannon Bowling, and Andreas Tolk at Old Dominion University for their endless patience, encouragement and valuable advice, without which I would never have completed this work. I am especially indebted to Colonel Steven Mains for many years of mentorship, friendship, and support. I would also like to thank Mr. Jeff Cares and Dr. Michael Bell for their generous support and cooperation.

Thank you all.

TABLE OF CONTENTS

Page

LIST OF TABLES VII

LIST OF FIGURES IX

INTRODUCTION 1

- BACKGROUND 1
- EVOLUTIONARY SHIFTS IN MODERN WARFARE 1
- CYBERWAR AND NETWAR 4
- INTO THE FOURTH GENERATION 5

LITERATURE REVIEW 7

- WHY ARE LEGACY COMBAT MODELS INADEQUATE? 7
- NETWORK CENTRIC OPERATIONS 8
- THE INFORMATION AGE COMBAT MODEL 12
- ANALYSIS OF THE IACM 17

METHODOLOGY 21

- AN AGENT-BASED SIMULATION MODEL USING THE IACM 21
- STRUCTURE OF THE EXPERIMENT 22
- DEVELOPING THE NETLOGO MODEL 29
- EXPERIMENTAL APPROACH 36

MODELING RESULTS 37

- MODELING A 7-3-7-1 NETWORK 37
- MODELING AN 8-3-8-1 NETWORK 41
- MODELING A 9-5-9-1 NETWORK 44

CONCLUSION 49

- ACHIEVING THE RESEARCH PURPOSE 49
- RECOMMENDATIONS FOR FUTURE WORK 50
- SUMMARY 51

REFERENCES 53

APPENDIX

- MEANINGFUL CONFIGURATIONS 56
- PERRON-FROBENIUS EIGENVALUE VALUES 75
- UNIQUE PERRON-FROBENIUS EIGENVALUE VALUES 81
- MODELING RESULTS 84
- REGRESSION RESULTS 97
- METRICS FOR A 7-3-7-1 NETWORK 103
- NETLOGO CODE 105

VITA 125

LIST OF TABLES

Page

Table 1. Types of links available in the IACM. 16

Table 2. The numbers of meaningfully different configurations of all X-Y-X-1 networked forces where X < 11 and Y < 8. 25

Table 3. The numbers of unique λ_{PFE} values of all X-Y-X-1 networked forces where X < 11 and Y < 8. 27

Table 4. The percentages of λ_{PFE} values that are unique for all X-Y-X-1 networked forces where X < 11 and Y < 8. 28

Table 5. All Possible Configurations of a 4-3-4-1 Network 56

Table 6. All Possible Configurations of a 5-3-5-1 Network 57

Table 7. All Possible Configurations of a 7-3-7-1 Network 58

Table 8. Meaningful Combinations of an X-3-X-1 Network 61

Table 9. Meaningful Combinations of an X-4-X-1 Network 66

Table 10. Meaningful Combinations of an X-5-X-1 Network 71

Table 11. Meaningful Combinations of an X-6-X-1 Network 73

Table 12. Meaningful Combinations of an X-7-X-1 Network 74

Table 13. λ_{PFE} Values for all Meaningful Configurations of a 7-3-7-1 Network 75

Table 14. λ_{PFE} Values for all Meaningful Configurations of a 8-3-8-1 Network 76

Table 15. λ_{PFE} Values for all Meaningful Configurations of a 9-5-9-1 Network 78

Table 16. Unique λ_{PFE} Values for all Meaningful Configurations of a 7-3-7-1 Network 81

Table 17. Unique λ_{PFE} Values for all Meaningful Configurations of an 8-3-8-1 Network 82

Table 18. Unique λ_{PFE} Values for all Meaningful Configurations of a 9-5-9-1 Network 83

Table 19. Modeling Results for a 7-3-7-1 Network 84

Table 20. Modeling Results for an 8-3-8-1 Network 85

Table 21. Modeling Results for a 9-5-9-1 Network 87

Table 22. Modeling Results for a 9-5-9-1 Network (not including ties) 90

Table 23. Modeling Results for a 9-5-9-1 Network (Ranked by λ_{PFE} value) 93

Table 24. Modeling Results for all 9-5-9-1 configurations with a λ_{PFE} value of 2.031 96

Table 25. Regression Results for a 7-3-7-1 Network 97

Table 26. Regression Results for an 8-3-8-1 Network 98

Table 27. Regression Results for a 9-5-9-1 Network 99

Table 28. Regression Results for a 9-5-9-1 Network (minus ties) 100

Table 29. Two-Variable Regression Results for a 9-5-9-1 Network (minus ties) 101

Table 30. Two-Variable Regression Results for a 9-5-9-1 Network 102

Table 31. Metric Values for a 7-3-7-1 Network 104

LIST OF FIGURES

Page

Figure 1. The basic combat network represents the simplest situation in which one side can influence another. 14

Figure 2. The simplest two-sided combat network consists of two opposing combat cycles. 14

Figure 3. The simplest complete combat network represents all the ways in which Sensors, Deciders, Influencers and Targets interact meaningfully with each other 15

Figure 4. An adjacency matrix for the simplest complete combat network 18

Figure 5. The two meaningfully different configurations of a 4-3-4-1 network 23

Figure 6. An adjacency matrix for one of the 42 meaningfully different configurations of a 7-3-7-1 network. 26

Figure 7. The probability of a BLUE Win for each of the 42 BLUE configurations (with λ_{BLUE} values varying from 1.821 to 2.280) against each of the 42 RED configurations (also with λ_{RED} values varying from 1.821 to 2.280). 38

Figure 8. The average probability of a BLUE win by λ_{PFE} for 42 configurations of a 7-3-7-1 BLUE network. 39

Figure 9. The disparity and robustness values of the 42 configurations of a 7-3-7-1 BLUE network. 40

Figure 10. The average probability of a BLUE win by λ_{PFE} for 78 configurations of an 8-3-8-1 BLUE network 42

Figure 11. The disparity and robustness values of the 78 configurations of an 8-3-8-1 BLUE network. 43

Figure 12. The average probability of a BLUE win by λ_{PFE} for 95 configurations of a 9-5-9-1 BLUE network. 44

Figure 13. The average probability of a BLUE win by λ_{PFE} for 95 configurations of a 9-5-9-1 BLUE network (not including ties). 46

Figure 14. The disparity and robustness values of the 95 configurations of a 9-5-9-1 BLUE network. 47

CHAPTER I

INTRODUCTION

"New approaches to accomplishing the functions that are associated with Command and Control are becoming an essential part of an Information Age transformation of military and civilian institutions; such a transformation is required to meet twenty-first century security challenges." (Alberts 2007, 2) [1]

BACKGROUND

While the nature of and the approach to command and control is evolving in order to meet the challenges of Information Age warfare, the essential functions that must be accomplished remain constant. One of those essential functions is the organization, or "arrangement" (from the definition of "Command and Control" in Joint Publication 1-02, 79-80)[2], of the assets within a combat force. Certainly, the many different ways to arrange a given set of assets will have different impacts on the combat effectiveness of the force. Some arrangements will enable self-synchronization, while other arrangements will impede it. How then, should an Information Age combat force be organized in order to optimize its effectiveness?

EVOLUTIONARY SHIFTS IN MODERN WARFARE

The organizations of military forces throughout history have been largely dependent on the capabilities of the weapons of that age. Lind, et al., (1989) provide the first classification of modern military development into distinct generations. The first generation of warfare consists of "the tactics of line and column" (Lind, et al. 1989, 23)

---

[1] Citation and reference list format for this manuscript are taken from *The International C2 Journal*.

[2] JP 1-02 defines *command and control* as "the exercise of authority and direction by a properly designated commander over assigned and attached forces in the accomplishment of the mission. Command and control functions are performed through an arrangement of personnel, equipment, communications, facilities, and

and is focused on massed manpower. Second generation warfare evolved due to the increased lethality of firearms and artillery and, coupled with the advent of the machinegun, indirect fire and other technological advances on the battlefield, resulted in a change of focus from massed manpower to massed firepower. The tactics of second generation warfare evolved towards fire and movement, although they remained linear in nature and focused on attrition. The third generation of warfare reflects the emergence of maneuver warfare and its employment on nonlinear tactics. This was manifested in the blitzkrieg style of war introduced in early World War II.

In addition to establishing classifications for the evolutionary steps of warfare, Lind, et al., (1989) identified the catalysts for change. The two predominant catalysts for change were technological advances and ideas. They attributed the initial shift towards the first generation to both of these catalysts, fueled further by the social conditions and changes of the French revolution. The principle cause for the shift towards second generation warfare was the advancing technology, but the emergence of the operational art of war represented an extraordinarily important idea to develop. The ideas of maneuver warfare were the primary drivers of change for the third generation.

Toffler (1993) proposes his own definitions of war based on three waves of societal evolution. He categorizes the transition between the ages of civilizations as waves, thereby implying their dynamic and expansive nature. The civilizations of the First Wave were the agrarian societies predominant from early history until the 18th century. Every aspect of these societies revolved around the land and man's use of it. Most of these societies were transformed by the Industrial Revolution. Toffler described the mass production capabilities of factories as a new way of creating wealth, and calls it the driving force behind many political, social and theological transformations. The intersection of the First and Second Wave societies inevitably resulted in conflicts, and Toffler credits the destabilizing effects of these transformations for being "the central tension from which other conflicts derived" (Toffler 1993, 20). Internal conflicts within each industrializing country were followed by the external conflicts of conquest, resulting

---

procedures employed by a commander in planning, directing, coordinating, and controlling forces and operations in the accomplishment of the mission."

in a "bisected" world of Second Wave civilizations dominating First Wave colonies (Toffler 1993, 21).

The emergence of the Information Age is the catalyst for the arrival of the Third Wave civilizations. A Third Wave civilization will dominate "based on the new ways in which it creates and exploits knowledge" (Toffler 1993, 22). Like its predecessors, the Third Wave's transformational impact is all-encompassing, destabilizing, and is further dividing the world. Toffler's summary of these three "contrasting and competing" civilizations is striking: "...the first still symbolized by the hoe; the second by the assembly line; and the third by the computer" (Toffler 1993, 21). He predicts that the resulting trisection of the world's societies will set "the context in which most wars from now on will be fought" (Toffler 1993, 25).

Toffler links the evolution of warfare to these three waves of civilizations, tracing each wave's way of war back to the catalyst that spawned it. Each wave is a reflection of the way its wealth is created and, subsequently, each wave's form of warfare reflects that source as well. Pre-industrial era armies were a refection of their agrarian roots, often campaigning only during the off-season and deriving their power from the feudal lords and landowners. The mass production capabilities of the Second Wave enabled mass production armies, resulting in higher technology weapons, mass conscription and the idea of a standing army for the nation-state. From the Napoleonic Wars through World War II, the ever-increasing efficiency of the Second Wave civilization resulted in the ever-increasing lethality of war, all focused on the destruction of the enemy armies or nations. Toffler argues that the arrival of Third Wave warfare is imminent as the "outer limits" of the range, speed and lethality of weapons has been reached (Toffler 1993, 43).

Toffler calls the Gulf War the advent of the Third Wave War. United States forces were tapping the awesome potential of information and the results were impressive. Increased knowledge allowed the American forces to accelerate their decisions and actions and bypass the traditional, linear approach to war. "Smart" weapons increased destructive efficiency and reduced collateral damage. Whereas Second Wave armies resemble "machines," Third Wave armies represent "thinking systems" (Toffler 1993, 80). The Gulf War represented the first clash between the Second and Third Waves and the battle was decided before it had begun.

CYBERWAR AND NETWAR

Arquilla and Ronfeldt (1993) also recognized that the nature of warfare was transforming, with information replacing hardware as the key commodity. They envisioned two new products of the information revolution: cyberwar and netwar. They define cyberwar as "conducting, and preparing to conduct, military operations according to information-related principles" (Arquilla and Ronfeldt 1993, 146). In essence, this term represents high-technology warfare where information is exploited in order to defeat an enemy's military capabilities. Netwar, on the other hand, reflects "information-related conflict at a grand level between nations or societies" (Arquilla and Ronfeldt 1993, 144). Such conflict may include non-state actors and will often be non-violent, although some low-intensity conflicts are included. Netwar employs all available networks to disrupt, damage, or modify what a target population thinks about itself and the world, as well as to influence public or elite opinion. Arquilla and Ronfeldt (1996; 1998; 2001) continued to develop the netwar concept by elaborating on its defining concept: the networked nature of the combatant's organization.

Arquilla and Ronfeldt (2000) investigated the concept of swarming as well, calling it the next evolutionary step in the nature of war. Early warfare was defined by the melee: large formations dissolving into small contests to resolve the battle. This was replaced by massing well-organized formations into set-piece battles. Massing was subsequently replaced by maneuver, which reflected synchronized mobile operations to disrupt the enemy before striking the decisive point with mass. Since the key weapon of netwar is information, then the key form of engagement must be able to exploit it. Swarming is that new form of engagement for netwar. They define swarming as "the systematic pulsing of force and/or fire by dispersed, internetted units, so as to strike the adversary from all directions simultaneously" (Arquilla and Ronfeldt 2000, 8). The key concept of swarming is "sustainable pulsing," which represents the swarmers' ability to disperse, concentrate to strike a common target from multiple directions, and then redisperse again. The swarmers are acting either autonomously or semi-autonomously with the intent of disrupting the enemy's cohesion. Swarming, therefore, is a characteristic of the next generation of war.

INTO THE FOURTH GENERATION

Lind, et al., (1989) identify four trends of the shifts between the first three generations of warfare that likely indicate what the characteristics of fourth generation warfare may be. The first trend indicates that fourth generation warfare will be "...widely dispersed and largely undefined; the distinction between war and peace will be blurred to the vanishing point" (Lind, et al. 1989, 23). It will be nonlinear, and "the distinction between 'civilian' and 'military' may disappear" (Lind, et al. 1989, 23). The second trend indicates that increases in dispersion and battlefield tempo will reduce the value of centralized logistics and a fixed military infrastructure. Thirdly, armies will forgo masses of men or firepower for increased maneuverability. Lastly, the desired endstate will change from the physical destruction of the enemy's armed forces to the elimination of the enemy's capabilities or will to fight. Lind, et al., (1989) concede that while these characteristics are already present to some degree in the third generation, they will become the defining characteristics in the next generation.

As previously established, the evolution to fourth generation warfare will be stimulated by some combination of the recent advances in technology and new ideas. Lind, et al. (1989) conducted an exploration into what a technology-driven fourth generation war might look like and established some additional characteristics. They suggest that a variety of futuristic weapons, such as directed energy weapons, robotics, and "smart" assets will be employed. The predominant theme amongst these visions is the expansion of war from the military domain into the political and cultural domains. Lind, et al., however, caveat this investigation with a skepticism that, while possible, such technological developments are not likely be fulfilled, or fulfilling.

Lind, et al., (1989) conduct a similar exploration into what an idea-driven fourth generation war might look like; in particular, a new generation driven by non-Western ideas originated from Islamic or Asiatic cultures. Concepts such as "collapsing the enemy" and turning an enemy's strength into a weakness originate with Sun Tzu. The disorderly nature of modern war provides an environment where terrorism may be effective. The lack of uniforms, rank, and order make it a perplexing challenge for a conventional western military approach. A combination of terrorism, high technology, and a transnational base is a good indication that the next generation of warfare has arrived.

Fourth generation warfare has impacted each of the three levels of war: strategic, operational, tactical. Where they were traditionally distinct from each other they are now intertwined, such that a tactical action can now have an immediate, significant strategic effect. The accelerating effect of television is a significant factor in this change. Lind (2005) points out that while an action can simultaneous impact each of the three levels of war, those impacts may not all be the same. They may even be contradictory. He uses overwhelming firepower as an example of this. While it may ensure tactical success in a particular situation, it most likely is counterproductive at the operational and strategic levels.

Hammes (1994, 2005) builds his definition of fourth generation warfare on the work of Lind, et al., (1989), while acknowledging Toffler (1993) and Van Creveld (2000) for providing valuable insight into the context of the problem. Hammes also embraces the concept of netwar by Arquilla and Ronfeldt (1993) as fourth generation warfare. He dismisses cyberwar, however, as technologically-oriented third generation warfare and expresses frustration with the Department of Defense's infatuation with it. Like his predecessors, he sees the next generation of warfare spanning the entire political, economic, social and technical spectrums. It will employ networks, and it will be lengthy. Since the evolution of warfare is continuous; the arrival of the fourth generation is not the end-state. Understanding fourth generation warfare is a necessary step in preparing for the fifth generation.

CHAPTER II

LITERATURE REVIEW

> "One of the great questions at the root of all strategy is that of *concentration*; the concentration of the whole resources of a belligerent on a single purpose or object, and concurrently the concentration of the main strength of his forces, whether naval or military, at one point in the filed of operations."
>
> - Frederick W. Lanchester (1914, 422)

WHY ARE LEGACY COMBAT MODELS INADEQUATE?

The concept of *concentration* is an important military tenet that was recognized as early as the 5th century B.C., by Sun Tzu. It remains relevant to the training of modern military forces in the form of the principles of war, specifically *mass*: to bring decisive force to bear at critical times and places. Concentration represents the essence of Lanchester's (1914) differential equations, which provided a method for calculating the rates of loss for two opposing forces. These equations assume a battlefield populated with homogenous, evenly-distributed entities, and are incapable of addressing the spatial and temporal complexities of modern combat. Epstein states that Lanchester's equations "offer a fundamentally implausible representation of combat under all but a very small set of circumstances" because they do not account for the ability of a force to withdraw from an engagement (i.e., "feedback"), nor do they allow for the "trading of space for time" (Epstein 1985, 4-6). Additionally, the casualty-exchange rates grow at a constant, not at a marginally diminishing rate.

Despite these shortcomings, Lanchester's equations became the basis of the mathematical calculations within many past and current combat models and simulations. Cares declares such models are inadequate representations of modern (post-Industrial Age) combat for the following reasons (Cares 2005, 33-34):

> "(1) The models rely on mathematics that represent combat activities as independent processes. Networked processes are by definition interdependent.

> (2) The models aggregate and disaggregate in a way that treats fine-scale behaviors as noise at the aggregate level. Such a process cannot adequately represent local tactical arrangements, clever use of information or massed effects from distributed forces. These, of course, are each important Information Age Warfare precepts.
>
> (3) The models do not reflect the fact that the distribution of networked performance is highly skewed. Network-enabled feedback and feed-forward mechanisms can create increasing returns – 'tipping point' behaviors – within Information Age systems. Although NCW [network centric warfare] concepts are said to capitalize on these types of networked effects, contemporary model assumptions deliberately inhibit them."

In summary, the nature of modern combat is nonlinear and Lanchester-based simulations are not able to adequately model it. Deterministic combat models, such as Lanchester's, preclude the types of non-linear outcomes possible in Information Age combat. Lanchester's equations can account for measures of performance such as lethality and weapon ranges; but they cannot account for command and control and networked effects. While Industrial Age combat may be defined by the former; Information Age combat is defined by the latter. Consequently, these legacy models may have been useful for modeling the large-scale, linear warfare of the Industrial Age, but they are not capable of providing insights into Information Age warfare.

NETWORK CENTRIC OPERATIONS

> "Network Centric Warfare (NCW) is no less than the embodiment of an Information Age transformation of the DoD. It involves a new way of thinking about how we accomplish our mission, how we organize and interrelate, and how we acquire and field the systems that support us."
>
> *Department of Defense Report to Congress* (Executive Summary, i)

The concept of Network Centric Operations[3] (NCO) represents a shift from traditional attrition-based approaches to a warfighting style that emphasizes speed of

[3] NCO was originally referred to as Network Centric Warfare (NCW)

command and self-synchronization. One goal of NCO is to field a force that is capable of achieving information superiority, i.e. "having a dramatically better awareness or understanding of the battlespace rather than simply more raw data" (Cebrowski and Garstka 1998, 32), thus enabling a massing of effects instead of the traditional massing of forces that will disrupt the enemy's strategy and preclude potential courses of action. Prior to the introduction of the NCO concept, assessment of the combat potential of a force tended to focus on force composition (the number of platforms or other entities of each type) and individual platform capabilities, with force lay-down (spatial distribution) and employment (tactics) as important but scenario-dependent factors. NCO shifts the focus towards the information-based aspects of force employment: information collection, communication, and exploitation. Central to the ability of a force to manage and exploit information is its connectivity: the existence, capacity, reliability, and other attributes of the links that connect its platforms, command and control centers, and other entities. A fundamental problem is to develop an understanding of the influence of connectivity on force effectiveness that can lead eventually to quantitative prediction and guidelines for design and employment. While NCO is reliant on the technological advances that will enable sensor, and data transfer and management capabilities to achieve the necessary speed of command, it is also dependent on the bottom-up organization that allows for self-synchronization. Cebrowski and Garstka state that in order "to fight on a network-centric rather than platform-centric basis, we must change how we train, how we organize, and how we *allocate* our resources" [emphasis added] (Cebrowski and Garstka 1998, 34).

Alberts, et al., (1999) expanded on this initial proposal by defining the characteristics of NCO and recommending a process for developing the desired NCO capabilities. They emphasized three key concepts of NCO: 1) military forces will be geographically dispersed, 2) these military forces will be empowered by knowledge, and 3) these military forces will be effectively linked. Alberts, et al., (2001) offered a more detailed articulation of information superiority and NCO. They achieved this by establishing a language that enables us to communicate and gain insight into the issues unique to Information Age warfare. This language is based on the three domains of warfare and the interactions between them: physical, information, and cognitive. While this insight yields a greater understanding of Information Age warfare, it does not provide a capability for a quantifiable value proposition, and no specific recommendations for organizational changes are made.

Alberts and Hayes (2003) stated again that current command and control concepts, organizations, and systems are inadequate, and warned against a fixation on technical solutions. They endorsed a *power to the edge* approach that empowers "individuals at the edge of an organization (where the organization interacts with its operating environment to have an impact or effect on that environment)" (Alberts and Hayes 2003, 5). *Power to the edge* achieved in each of the domains of warfare (physical, information, cognitive) will result in a self-synchronizing capability. This self-synchronizing capability is dependent on certain assumptions (Alberts and Hayes 2003, 27):

- "Clear and consistent understanding of command intent;
- High quality information and shared situational awareness;
- Competence at all levels of the force; and
- Trust in the information, subordinates, superiors, peers, and equipment."

While *Power to the Edge* by Alberts and Hayes (2003) reflects a continued refinement of the concepts and theory of NCO, it does not offer any new techniques or metrics for quantifying network performance. Alberts, et al., (1999) acknowledge that the NCO theory alone cannot provide what we are looking for: "As such, we cannot simply apply new technologies to the current platforms, organizations, and doctrine of warfare" (Alberts, et al. 1999, 3). If this is so, how do we organize?

Cebrowski and Garstka (1998) did include one quantifiable metric. They identified that network-centric computing "is governed by Metcalf's Law, which asserts that the 'power' of a network is proportional to the square of the number of nodes in the network" (Cebrowski and Garstka 1998, 30). Consequently, more nodes equal more "power". They concluded that we should pursue the same value proposition in preparation for network centric warfare. Alberts, et al., (1999) also employ Metcalf's Law to quantify the power of a network, but they do acknowledge that it is merely a measurement of potential gains which will not be realized without "appropriate organizational and doctrinal changes" (Alberts, et al. 1999, 103). It is a measurement of potential because a straight application of Metcalf's Law assumes that all interactions have an equal positive value, and this is an assumption that Alberts, et al., explicitly reject. Consequently, they employ Metcalf's Law to calculate a network's potential for information interaction, but recognize that further application of this quantification would require "a value-creation logic and a user-defined value preference (value function)" (Alberts, et al. 1999, 264). They do not, however, propose any other metrics for quantifying the performance of a network.

Alberts, et al., (2001) do propose to measure network performance according to the attributes of *information richness*, i.e., "an aggregate measure of the quality of information (Alberts, et al. 2001, 46), and *information reach*, i.e., "an aggregate measure of the degree that information is shared" (Alberts, et al. 2001, 46); but, unfortunately, these metrics are predominantly either information technology metrics (i.e., measurements of hardware, software, and computing network performance) or traditional platform performance metrics (i.e., lethality, survivability, etc.). None of the metrics used in the examples directly quantify organizational or doctrinal attributes. They do, however, introduce a quantifiable metric for measuring the degree of synchronization, i.e., an "output characteristic of the C2[4] processes that arrange and continually adapt the relationships of actions (including moving and tasking forces) in time and space in order to achieve the established objective(s)," (Alberts, et al. 2001, 206). Synchronization is an important military principle that is becoming increasingly difficult to achieve with the growing speed and complexity of warfare. To measure synchronization, S, each interaction between every entity in the network is assigned a value between +1 and -1, which reflects the degree of synchronization (+1, if perfect) or interference (-1, if perfect), or neutrality (0) of the relationship between that particular pair. These values are then summed up in a combinatorial manner. The value of S can be measured over time, thereby allowing us to calculate and compare the rates of change of S between different networks. While Alberts, et al., (2001) admit that this metric will likely need refining, it is a useful step towards quantifying network performance.

Ling, et al., (2005) agree that the body of knowledge lacks effective quantifiable metrics with which to measure network performance: "Specifically, there is currently no

---

[4] C2 is an acronym for "command and control".

clear means by which one can link the internal metrics of the performance of a network to the external measure of the decision-action cycle rate for a networked force" (Ling, et al. 2005, 5). While their refinements of the metrics of connectivity, reach, richness and tempo represent another important developmental step, the utility of these metrics in measuring force effectiveness is not yet known. More recently, Reid, et al. (2007) reiterated the lack of quantifiable metrics in NCO theory: "It predicts no limits to the useful capacity of the network, and so cannot answer our questions about how to make effective use of modest resources. How are we to make difficult balance-of-investment decisions in light of such advice?" (Reid, et al. 2007, 337). In other words, the theory of NCO remains incapable of providing a value proposition for organizing a networked combat force.

THE INFORMATION AGE COMBAT MODEL

The Information Age Combat Model (IACM), recently introduced by Cares (2005), attempts to describe combat (or competition) between distributed, networked forces or organizations. The basic objects of this model are not platforms or other entities capable of independent action, but rather nodes that can perform elementary tasks (sense, decide, or influence) and links that connect these nodes. Information flow between the nodes is generally necessary for any useful activity to occur. This focus on "network-centric" rather than "platform-centric" operations is intended to advance the state of the art in combat modeling "by explicitly representing interdependencies, properly representing complex local behaviors and capturing the skewed distribution of networked performance" (Cares 2005, 34). The content of this section is mostly included in Deller, et al. (2009).

The IACM employs four types of nodes defined by the following properties:

- *Sensors* receive signals about observable phenomena from other nodes and send them to Deciders;
- *Deciders* receive information from Sensors and make decisions about the present and future arrangements of other nodes;

- *Influencers* receive directions from Deciders and interact with other nodes to affect the state of those nodes;
- *Targets* are nodes that have military value but are not Sensors, Deciders, or Influencers.

These properties represent the minimum required for each type of node. Other possible characteristics will emerge in the following discussion. Each node belongs to a "side" in the competition, of which there are at least two. We will restrict the present discussion to two sides, conventionally termed BLUE (depicted in black in the figures) and RED (depicted in gray). In principle, any pair of nodes can interact, regardless of side, but some restrictions will be found to occur for both theoretical and practical reasons. It is worth noting that Influencers can act on any type of node, and Sensors can detect any type. The Target type was introduced primarily to reflect the fact that not all military assets fall into one of the other three types. In most situations, however, an Influencer will target an adversary Sensor, Decider, or Influencer.

The basic combat network shown in Figure 1 represents the simplest situation in which one side can influence another. The BLUE Sensor (S) detects the RED Target (T) and informs the BLUE Decider (D) of the contact. The Decider then instructs the BLUE influencer (I) to engage the Target. The Influencer initiates effects, such as exerting physical force, psychological or social influence, or other forms of influence on the target. The process may be repeated until the Decider determines that the desired effect has been achieved. It should be noted that the effect assessment requires sensing, which means that this will be conducted in a new circle. This most basic combat network is also referred to as a *combat cycle*.

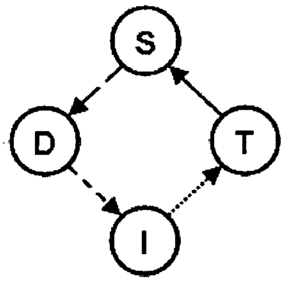

Figure 1. The basic combat network represents the simplest situation in which one side can influence another.

Each of the four links in Figure 1 is shown with a different type of line in order to emphasize the fact that the flows across these links may be very different. In particular, some links may represent purely physical interactions, while others may entail both physical processes and information flows. Two opposing combat cycles comprise the simplest two-sided combat network (Figure 2). The figures in this dissertation utilize the basic elements of graph theory. For more details on graph theory the interested reader is referred to Chartrand (1984).

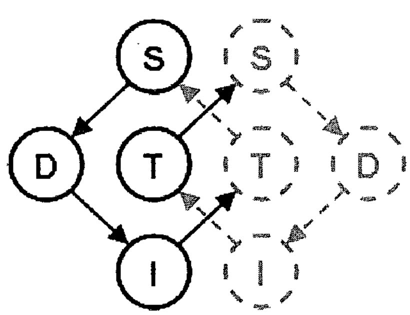

Figure 2. The simplest two-sided combat network consists of two opposing combat cycles.

Cares (2005) described the simplest complete (two-sided) combat network as having 36 possible links. While the number of possible links for eight nodes (four each for BLUE and RED) is 64, we were able to exclude 28 and reduce that number to 36 based on the following important assumptions. The results are shown in Figure 3.

- Targets are passive; their only role is to be sensed and influenced. Therefore, 12 links from Targets to any nodes other than a Sensor were excluded.
- Sensors take no action; they provide information to Deciders and Sensors. Therefore, 10 links from Sensors to any nodes other than a Sensor or own Decider were excluded.
- Deciders act only through Influencers but can be sensed. Therefore, 6 links from Deciders to any adversary nodes except a Sensor were excluded.

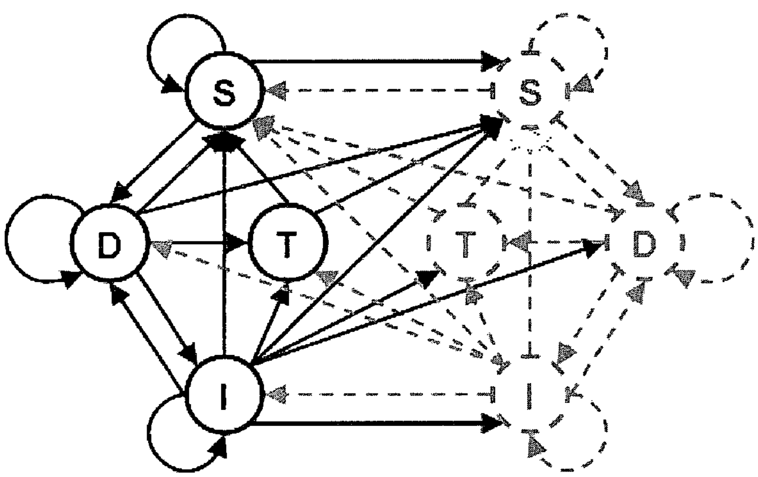

Figure 3. The simplest complete combat network represents all the ways in which Sensors, Deciders, Influencers and Targets interact meaningfully with each other.

When the BLUE/RED symmetry is taken into account, the number of link types is reduced to 18. These are listed in Table 1, where the nodes are identified as in Figure 3. Links between a node and itself in Figure 3 have been interpreted as connecting two different nodes of the same type and side.

Table 1. Types of links available in the IACM.

Link Type	From	To	Interpretation	Link Type	From	To	Interpretation
1	S_{BLUE} S_{RED}	S_{BLUE} S_{RED}	S detecting own S, or S coordinating with own S	10	I_{BLUE} I_{RED}	D_{BLUE} D_{RED}	I attacking own D, or I reporting to own D
2	S_{BLUE} S_{RED}	D_{BLUE} D_{RED}	S reporting to own D	11	I_{BLUE} I_{RED}	I_{BLUE} I_{RED}	I attacking own I, or I coordinating with own I
3	S_{BLUE} S_{RED}	S_{RED} S_{BLUE}	S detecting adversary S	12	I_{BLUE} I_{RED}	T_{BLUE} T_{RED}	I attacking own T
4	D_{BLUE} D_{RED}	S_{BLUE} S_{RED}	S detecting own D, or D commanding own S	13	I_{BLUE} I_{RED}	S_{RED} S_{BLUE}	I attacking adversary S, or S detecting adversary I
5	D_{BLUE} D_{RED}	D_{BLUE} D_{RED}	D commanding own D	14	I_{BLUE} I_{RED}	D_{RED} D_{BLUE}	I attacking adversary D
6	D_{BLUE} D_{RED}	I_{BLUE} I_{RED}	D commanding own I	15	I_{BLUE} I_{RED}	I_{RED} I_{BLUE}	I attacking adversary I
7	D_{BLUE} D_{RED}	T_{BLUE} T_{RED}	D commanding own T	16	I_{BLUE} I_{RED}	T_{RED} T_{BLUE}	I attacking adversary T
8	D_{BLUE} D_{RED}	S_{RED} S_{BLUE}	S detecting adversary D	17	T_{BLUE} T_{RED}	S_{BLUE} S_{RED}	S detecting own T
9	I_{BLUE} I_{RED}	S_{BLUE} S_{RED}	I attacking own S, or S detecting own I	18	T_{BLUE} T_{RED}	S_{RED} S_{BLUE}	S detecting adversary T

The interpretation of some of the links (types 1, 4, 9, 10, 11, and 13 in Table 1) is ambiguous. This was recognized in the initial development of the IACM (Cares 2005), and resolving this issue was described as both "the next major advance" in the development of the model and a requirement for "practical" (i.e., quantitative) analysis based on it. The simulations presented here are a step in this direction, since they employ only basic combat networks similar to Figure 1, but with the Target replaced by an adversary Sensor or Influencer. These combat cycles (Cares 2005) contain only links of types 2, 3, 6, 13, and 15. Of these, only type 13 is ambiguous. Both interpretations of this link will be used, but the context of the model always makes clear which is intended.

ANALYSIS OF THE IACM

Once the IACM has been defined in terms of a network of nodes and links, the language and tools of graph theory (see, for example, Chartrand 1984) can be used for both description and analysis. A concise description of any graph is provided by the adjacency matrix A, in which the row and column indices represent the nodes, and the matrix elements are either one or zero according to the rule: $A_{ij} = 1$, if there exists a link from node i to node j and $A_{ij} = 0$, otherwise. Many properties of a graph or network can be calculated directly from the adjacency matrix, and two are of particular interest here. Since combat power or influence can be exerted only when there exists a connected cycle that includes the node to be influenced, the detection of cycles in the graph is of great importance. Cares attributes the increased capabilities of networked forces to the presence of *cycles*, which are represented by sub-networks within the overall combat network. These sub-networks are arrangements of linked nodes where the path of directional links revisits at least one node previously departed from. The absence of cycles indicates the absence of any potential networked effects. Cares defines four different types of cycles in his model (*control cycles*, *catalytic control cycles*, *catalytic competitive cycles*, and *combat cycles*) which reflect the ability to direct actions, share information, and influence enemy nodes. The content of this section is mostly included in Deller, et al. (2009).

One method used in studying the evolution of complex adaptive systems (chemical, biological, social, and economic) is calculation of the principal (maximum) eigenvalue of the adjacency matrix (Jain and Krishna 1998). The existence of a real, positive principal eigenvalue of A_{ij} is guaranteed by the Perron-Frobenius theorem, and this eigenvalue λ_{PFE} (and the corresponding eigenvector) are often referred to as the Perron-Frobenius Eigenvalue (eigenvector). It is readily shown (Jain and Krishna, 1999) that for a graph with no closed cycles $\lambda_{PFE} = 0$. For a graph with a single cycle of any length, one obtains $\lambda_{PFE} = 1$. Graphs with more complicated cycle structures have $\lambda_{PFE} > 1$. This had led to the proposal (Cares 2005) that λ_{PFE} be adopted as a measure of the ability of a network to produce feedback effects in general and combat power specifically in the case of the IACM. This is essentially the hypothesis explored in the present work.

Note that the links in both the graphical depiction and the adjacency matrix are directional, in that their meanings differ depending on which nodes they go "from" (left column) and "to" (top row). Figure 4 is an adjacency matrix representation of the simplest complete combat network depicted in Figure 3. Given that the number of different sub-networks for any N x N matrix is $2^{(N \times N)}$, it is obvious that attempts to optimize the arrangement of nodes and links for any but the simplest combat networks quickly becomes impossible.

$$
\begin{array}{cc}
 & \textit{To} \\
 & \begin{array}{cccccccc} S & D & I & T & S & D & I & T \end{array} \\
\textit{From} \begin{array}{c} S \\ D \\ I \\ T \\ S \\ D \\ I \\ T \end{array} &
\left[\begin{array}{cccccccc}
1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{array}\right]
\end{array}
$$

Figure 4. An adjacency matrix for the simplest complete combat network.

An alternative, but closely related approach is based on the fact that A^n (the n^{th} power of the adjacency matrix) can be used to obtain the number of distinct paths connecting any pair of nodes (Chartrand 1984). Specifically, $(A^n)_{ij}$, which is the ij matrix element of A^n, is equal to the number of distinct paths of length n connecting nodes i and j. In particular, $(A^n)_{ii}$ is the number of distinct closed paths from node i back to itself. If node i is an adversary Target T, then $(A^n)_{TT}$ is the number of distinct combat cycles of length n that include T. This represents the number of different ways that T can be engaged by the opposing force. In general, combat cycles must be of length at least four, and if the links are restricted to the types shown in Figure 1, they must be of length exactly four. In this case, the matrix element $(A^4)_{TT}$ is equal to the number of combat cycles that pass through the Target T and is therefore a potential measure of the combat power that can be brought to bear on it. Under special conditions that will be described below, it is possible to establish a quantitative connection between λ_{PFE} and $(A^4)_{TT}$, lending further support to the hypothesis that λ_{PFE} can be used as a measure of effectiveness for a distributed, networked force or organization.

It is important to note that the IACM does not rely on Metcalf's Law to measure the power of a network. While the λ_{PFE} is the predominate metric, Cares (2005) compiled a number of other statistics from numerous disciplines to measure the adaptability, robustness, survivability and other properties of a network:

- *Number of nodes* (N).
- *Link to node ratio* (l/N): This ratio will vary depending on whether the network is *minimally connected*, where all the nodes within the network are connected with the minimum number of links possible (i.e., $l = N - 1$), or *maximally connected*, where every node is directly connected to every other node (i.e., $l = (N - 1)!$). Cares claims that "very good connectivity can be achieved with orders of magnitude fewer links than the NCW literature suggests" (Cares 2005, 102).
- *Degree distribution*: A measurement of whether the number of links connected to each node is uniformly distributed throughout a network. Adaptive, complex networks have a skew degree distribution (i.e., a very small number of highly connected nodes).
- *Size, connectivity of largest hubs*: In order to reduce the network's Robustness, the largest hubs should not be connected to each other.
- *Characteristic path length:* The median of the mean of the lengths of all the shortest paths in the network.

- *Clustering coefficient*: The average clustering coefficient is a measurement of a network's cohesion and self-synchronization, and is calculated from the proportion of a node's direct neighbors that are also direct neighbors of each other. The best type of autocatalytic cycles in a combat network are 3-node cycles because they represent feedback or feed-forward shortcuts in the 4-node combat cycles. The distribution of clustering coefficients among all nodes should be skewed.
- *Betweenness*: The measure of the proportion of shortest paths that pass through a node (i.e., a node's importance to the network structure). Combat networks should have a skew distribution of betweenness.
- *Path horizon:* The measure of the number of nodes on average that a node must interact with for consecutive self-synchronization to occur. Self-synchronization occurs when the path horizon is the logarithm of the number of nodes.
- *Neutrality rating* ($(l - N + 1)/N$): A measure of the additional structure in a complex network over and above the minimum connectivity requirements.
- *Coefficient of networked effects* (λ_{PFE}/N): The CNE is a measure of the amount of cyclic behavior per node, and is calculated by normalizing the adjacency matrix's λ_{PFE}.
- *Susceptibility:* A measure of the number of links or nodes that can be removed before dynamic structure begins to break down.

For a more detailed understanding of these rules of thumb or the Information Age Combat Model see Cares (2005).

CHAPTER III

METHODOLOGY

AN AGENT-BASED SIMULATION MODEL USING THE IACM

The structure of the IACM makes it clear that the λ_{PFE} is a quantifiable metric with which to measure the organization of a networked force, but is it an indicator of combat effectiveness? To determine this we constructed an agent-based simulation representation of the IACM and conducted a series of force-on-force engagements using opposing forces of equal assets and capabilities, but differing in their connectivity arrangements or configurations. These differences in connectivity often, but not necessarily, lead to unequal λ_{PFE} values.

The agent-based paradigm was utilized for this purpose because the resulting models provide the ability to account for small unit organization, maneuver, and the networked effects that are the focus of our investigation. An additional advantage of utilizing an agent-based simulation was the ability to work around the ambiguities of link interpretation in the IACM. For example, instead of a mutually exclusive choice between defining a directional link from a BLUE Influencer to a RED Sensor (type 13 in Table 1) as either the Influencer "targeting" the Sensor or as the Sensor "sensing" the Influencer, both abilities can be represented in the agent-based simulation.

The first challenge in modeling the IACM concerned the adjacency matrix representation of the network. The IACM as originally described by Cares (2005) uses a single adjacency matrix to reflect the collective organization of both BLUE and RED forces. In this approach, the λ_{PFE} value is dependent on the configurations of both the BLUE and RED forces and might well represent the extent to which feedback effects occur in the engagement. Obviously, BLUE and RED each seek separately to maximize their own networked effects while minimizing those of the opposing force. This cannot be represented by a single λ_{PFE} value, so we calculate separate values (λ_{BLUE} and λ_{RED}) to reflect the potential networked effects of the configurations of each of the opposing forces. These calculations required the adjacency matrices include a single Target node representative of all the enemy forces capable of being targeted. In other words, the values of λ_{BLUE} and λ_{RED} are determined solely by the arrangement of their respective assets, independent of the asset arrangement of the opposing force.

STRUCTURE OF THE EXPERIMENT

In order to best associate any difference in force effectiveness to the difference in connectivity, the opposing forces consisted of the same number of Sensors, Deciders, and Influencers, differing only in the manner in which they were arranged (i.e., linked). Since the potential value of a Sensor may not equal the potential value of an Influencer, the composition of each configuration considered in this work contained an equal number Sensors and Influencer to preclude any bias towards those configurations that have more of one or the other. Additionally, both types of nodes had identical performance capabilities (i.e., the sensing range was chosen equal to the influencing range, and the speeds of movement of the two types of node were equal). Consequently, the composition of both forces followed an X-Y-X-1 (Sensor-Decider-Influencer-Target) template.

For any particular values of X and Y, there is a finite number of ways to arrange those assets. In order to gain a "first order" understanding of the IACM, we made two key scoping decisions. First, each Sensor and Influencer would only be connected to one Decider (but any given Decider could be connected to multiple Sensors and Influencers). Second, the connectivity within any X-Y-X-1 force was limited to only those "vertical" links necessary to create combat cycles (i.e., link types 2, 3, 6, 13, and 15 in Table 1), which are the essence of the λ_{PFE} (the most basic element of the IACM). As noted below, future work will include "horizontal" links between Sensors, Deciders, and Influencers, such as link types 1, 5, and others. This can significantly enhance both the λ_{PFE} value and the performance of any given network configuration, but it requires the introduction of additional rules to manage and exploit the information carried by these links. The present model provides a baseline for assessing the effect of adding additional types of links. In addition the simplification used here is sufficient to insure that $(\lambda_{PFE})^4$ is equal to $(A^4)_{TT}$, providing the exact, quantitative relationship mentioned previously between λ_{PFE} and the number of combat cycles.

While the X-Y-X-1 template significantly scoped the focus of this effort, the number of possible configurations for a given force still becomes large very quickly. For example, there are a total of nine possible ways to distribute four Sensors and four Influencers across three Deciders (see Table 5 in the Appendix). No matter how you distribute them, one Decider will have two Sensors linked to it, and one Decider (which may or may not be the same Decider) will have two Influencers assigned to it. Fortunately, since the nodes of the IACM are generic it is possible to reduce this set by eliminating those configurations that are, in effect, identical. The only *meaningful* difference between the nine possible configurations of a 4-3-4-1 networked force is whether the Decider that is linked to two Sensors is the same Decider that is linked to two Influencers (see Figure 5). The remaining seven possible configurations are all modeled identically to these two configurations in the IACM (and are shaded gray in Table 5).

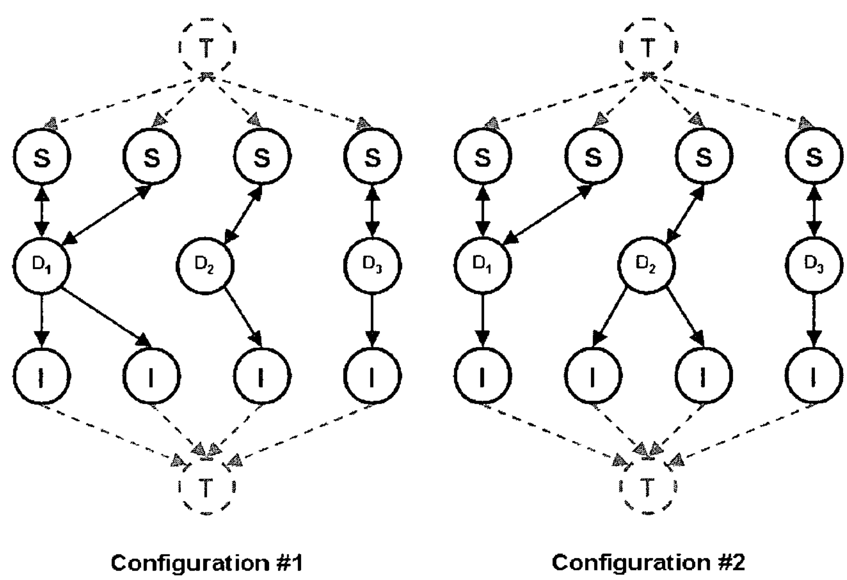

Figure 5. The two meaningfully different configurations of a 4-3-4-1 network.

Adding a single Sensor and Influencer yields a 5-3-5-1 networked force, which can be organized in 36 different ways (see Table 6). By applying this same logic, we reduce those 36 possible configurations to only eight meaningfully different configurations. Even with these most basic of examples, the difference between the number of possible configurations and number of meaningfully different configurations becomes quite apparent.

Identifying the meaningfully different configurations is crucial for the purpose of scoping the problem. While a 7-3-7-1 networked has 225 possible configurations (see Table 7), applying this same logic reduces this to a much more manageable number of only 42 meaningfully different configurations. Testing each of the 225 possible configurations of a 7-3-7-1 networked force against all 225 possible configurations of an opposing 7-3-7-1 networked force would require 50,625 (i.e., 225^2) unique engagements, but 42 combinations would only require 1,764 (i.e., 42^2) unique engagements. Since the number of meaningfully different combinations for any given set of nodes is a function of the number of unique values of the allocation combinations of X across Y, we attempted to define the function in order to automatically generate the combinations. This was not a simple task. Although the allocation resembles a partition problem, the exact numerical sequence of the numbers of meaningful combinations was difficult to establish. Since determining what this function might be is not the purpose of this research, we calculated the numbers of meaningfully different configurations for all X-Y-X-1 forces where $X < 11$ and $Y < 8$ using a simple algorithm based on the numbers of unique values for the distributions of Sensors and Influencers across the Deciders. The resulting totals are summarized in Table 2:

Table 2. The numbers of meaningfully different configurations of all X-Y-X-1 networked forces where X < 11 and Y < 8.

Numbers of Sensors (X) and Influencers (X)	Number of Deciders (Y): 3	4	5	6	7
3	1				
4	2	1			
5	8	2	1		
6	19	9	2	1	
7	42	27	9	2	1
8	78	74	30	9	2
9	139	168	95	31	9
10	224	363	248	105	31

Each of these configurations has a unique adjacency matrix representative of the connectivity of its nodes. The adjacency matrices for all configurations will only differ in two sections (see the unshaded sections of an example adjacency matrix in Figure 6), regardless of the total numbers of Sensors, Deciders, or Influencers. These unshaded sections reflect the connectivity of each Sensor and Influencer to and from a particular Decider, and vary by configuration based on the allocation of Sensors and Influencers across the Deciders. The shaded areas represent the absolute absence of any links between those types of nodes (such as the "horizontal" links discussed earlier), or the absolute existence of links between those types of nodes (such as the links from all BLUE influencers to the RED Target). Since fourteen of sixteen sections of the adjacency matrices for each of the 42 configurations are identical, the variance between the λ_{PFE} values is greatly reduced.

To

From	S	S	S	S	S	S	S	D	D	D	I	I	I	I	I	I	I	T
S	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
T	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Figure 6. An adjacency matrix for one of the 42 meaningfully different configurations of a 7-3-7-1 network.

In the case of a 7-3-7-1 networked force, the 42 meaningfully different configurations had 30 unique λ_{PFE} values ranging from 1.821 to 2.280 (see Table 13 in the Appendix). The λ_{PFE} values were calculated using this matrix calculator: http://www.bluebit.gr/matrix-calculator/. The adjacency matrix depicted in Figure 6 served as a template by varying the links in the two unshaded regions to represent all 42 meaningfully different combinations. The particular configuration used in this example represents Configuration ID #0[5] which has a λ_{PFE} value of 2.280. The λ_{PFE} values for each of the meaningfully different combinations of both an 8-3-8-1 and 9-5-9-1 networked forces are listed in Tables 14 and 15. Tables 16, 17, and 18 detail the unique λ_{PFE} values for the meaningfully different combinations of a 7-3-7-1, an 8-3-8-1, and a 9-5-9-1 networked forces.

[5] Each meaningful configuration is identified by a number. These ID numbers started at "0" due to the code used in the Behavior Space experimentation feature of the NetLogo agent-based model used in this research.

Identical configurations always have the same λ_{PFE} value; however, it is possible for meaningfully different configurations to share the same λ_{PFE} value. When this occurs, the λ_{PFE} loses its utility as an indicator of potential performance between these configurations. The numbers of unique λ_{PFE} values for the meaningful configurations for all X-Y-X-1 forces where X < 11 and Y < 8 are depicted in Table 3:

Table 3. The numbers of unique λ_{PFE} values of all X-Y-X-1 networked forces where X < 11 and Y < 8.

		Number of Deciders (Y)				
		3	**4**	**5**	**6**	**7**
Numbers of Sensors (X) and Influencers (X)	**3**	1				
	4	2	1			
	5	4	2	1		
	6	8	4	2	1	
	7	13	8	4	2	1
	8	20	13	8	4	2
	9	27	20	13	8	4
	10	38	27	20	13	8

Note that the numbers of unique λ_{PFE} values are not directly proportional to the numbers of meaningfully different configurations. For example, while an 8-3-8-1 networked force has 78 meaningfully different configurations with 20 unique λ_{PFE} values, the 95 meaningfully different configurations of a 9-5-9-1 networked force only have 13 unique λ_{PFE} values. This reduction will have a significant impact on the analysis of the modeling results presented later in this research. Table 4 depicts the percentage of λ_{PFE} values that are unique for the meaningful configurations for all X-Y-X-1 forces where X < 11 and Y < 8:

Table 4. The percentages of λ_{PFE} values that are unique for all X-Y-X-1 networked forces where X < 11 and Y < 8.

Numbers of Sensors (X) and Influencers (X)	Number of Deciders (Y): 3	4	5	6	7
3	100%				
4	100%	100%			
5	50.00%	100%	100%		
6	42.11%	44.44%	100%	100%	
7	30.95%	29.63%	44.44%	100%	100%
8	25.64%	17.57%	26.67%	44.44%	100%
9	19.42%	11.90%	13.68%	25.81%	44.44%
10	16.96%	7.44%	8.06%	12.38%	25.81%

The full range of mathematical values for a λ_{PFE} of an adjacency matrix containing 18 nodes is from 0 (for a network with no links at all) to 18 (for a maximally connected network). Note that the range of λ_{PFE} values for the 42 meaningfully different combinations of a 7-3-7-1 force is only a small segment (1.821 to 2.280) of the full range of possible values due to the relatively small differences of the links within any two of those configurations. The cause of this constrained range, however, is significant and becomes apparent when applying the statistical measures from various studies of network systems compiled by Cares (2005).

Of all the network statistics referenced in those studies, the λ_{PFE} (and its dependent Coefficient of Networked Effects) was the only one that varied in value between the 42 meaningfully different configurations; all others remained constant (see Table 31). Each configuration consists of the same numbers of nodes (18) and links (28). Each configuration shares a similar skewed degree distribution and, consequently, since each path within these configurations is the shortest path, the betweeness value for each configuration must also be skewed. Each configuration lacks any direct connectivity between its largest hubs, such as between Deciders or from any Decider directly to the

Target node. Likewise, the characteristic path length, clustering coefficient, path horizon, neutrality rating, and susceptibility of each of these configurations is identical. Given that the λ_{PFE} is the only one of these metrics that varies between these configurations, it is the only one of these metrics that might measure any potential variation in the effectiveness of these 42 configurations.

DEVELOPING THE NETLOGO MODEL

The agent-based simulation environment utilized for this research was NetLogo (Wilenski 1999). The purpose of this section is to explain the underlying logic of key parts of the NetLogo code utilized in this research; the entire code is provided in the Appendix). The code of the agent-based model closely follows the logic of the IACM, with a few notable exceptions. Agents served as Sensors, Deciders, and Influencers, but Targets were not included as they served no purpose other than to absorb losses. Given that this work represents a "first cut" effort, including Target agents with no detect, direct, or influence capabilities would only serve to clutter the results.

Additionally, Deciders cannot be destroyed in the present model. This was done in recognition of their unique role in connecting multiple Sensors and Influencers. Destruction of a Decider typically renders a number of other nodes useless (effectively destroyed), making it a particularly high value target. Since targets are detected and engaged in random order in our model, we wished to give all targets equal value in order not to generate atypical engagements that might bias the results.

The agent rules sets, themselves, function in accordance with the IACM. Sensors detect enemy nodes within the sensing range parameter, and communicate that information to their assigned (connected) Deciders. Deciders communicate the sensing information to their assigned Influencers. Influencers destroy the nearest enemy node that is both "sensed" by a Sensor connected to that Influencer's Decider, and within the influencing range parameter. Deciders direct Sensor movement towards areas of suspected enemy nodes. Deciders direct Influencers to move towards the nearest "sensed" enemy node. All nodes are assumed to perform their functions perfectly and instantaneously. Agent interactions are deterministic, i.e., the probabilities of detect, communicate and kill are all "1". A stochastic dimension to the model can be built once a better understanding of the research questions is gained, and this new dimension can be used to model errors and delays representing technological and human performance factors. Most importantly, the rules sets and parameter values for both BLUE and RED agents were identical.

Each agent in the model is defined as a part of an agentset (i.e., "breed") associated with a particular Decider. Since the nodes of the IACM are generic, the most important defining characteristic of any agent is its connectivity. For example, all BLUE Sensors and Influencers connected to the BLUE Decider_1 are established by the following breeds:

```
breed [ BInfluencer1s BInfluencer1 ]
breed [ BSensor1s BSensor1 ]
```

The actual numbers of agents within these breeds will vary according to the configuration being tested. Sliders were utilized for this purpose, thereby enabling the BehaviorSpace feature to vary the configurations automatically. The BLUE Decider_1 itself is also defined as a breed, but consists only of just that single agent:

```
breed [ BDecider1s BDecider1 ]
```

Similar agents for all other BLUE and RED Sensors, Influencers and Deciders were established.

The connectivity between these breeds represents the combat cycle links of the IACM (specifically link types 2, 3, 6, 13 and 15 as explained in Table 1). Link types 2 ("detection"), 6 ("order"), 13 ("LOF")[6] and 15 ("LOF") are defined in the simulation by the "directed-link-breed" keyword:

```
directed-link-breed [detections detection]
directed-link-breed [orders order]
directed-link-breed [LOFs LOF]
```

As mentioned earlier in this research, link type 13 has an ambiguous meaning in the IACM. The "directed-link-breed" keyword defines the Influencer-to-Sensor link as the Influencer attacking an enemy Sensor. Both link type 3 and the other IACM interpretation of link type 13 (i.e., a Sensor detecting an adversary Influencer) will be defined by the "sense" procedure later in the code. Finally, all agents within each breed

[6] LOF is an acronym for "line of fire," which is a direct horizontal line from a firing weapon to its target.

have certain variables that are tracked during the simulation, such as "side" (i.e., BLUE or RED), "dead" (i.e., agents that are attacked by an opposing Influencer may no longer act), and "sensed" (i.e., at any given "tick" count within the simulation an agent may be within sensing range of one or more opposing Sensors).

Given the large number of engagements within this experiment, it was imperative to utilize the BehaviorSpace feature of NetLogo. To enable this, each of the meaningfully different configurations was defined by using the "set" command to establish the appropriate numbers of Sensors and Influencers for each of the BLUE and RED Deciders. For example, BLUE Configuration (i.e., "BID") #0 for a 9-5-9-1 networked force assigned 5 Sensors and 5 Influencers to BLUE Decider_1, and one of each to the other 4 Deciders:

```
if BID = 0 [set Bconfig [5 1 1 1 1 5 1 1 1 1]]
if BID = 2 [set Bconfig [5 1 1 1 1 4 2 1 1 1]]
  set number-BSensor1s item 0 Bconfig
  set number-BSensor2s item 1 Bconfig
  set number-BSensor3s item 2 Bconfig
  set number-BSensor4s item 3 Bconfig
  set number-BSensor5s item 4 Bconfig
  set number-BInfluencer1s item 5 Bconfig
  set number-BInfluencer2s item 6 Bconfig
  set number-BInfluencer3s item 7 Bconfig
  set number-BInfluencer4s item 8 Bconfig
  set number-BInfluencer5s item 9 Bconfig
```

BLUE Configuration #2 is nearly identical, differing only in one link. Decider_1 now only has 4 assigned Influencers while Decider_2 now has 2. The movement of a single link is not trivial as it may have a significant impact on both the λ_{PFE} value and the average probability of Win for that particular configuration. Establishing all 95 meaningfully different configurations of a 9-5-9-1 networked force in this manner allows the BehaviorSpace feature to automatically cycle through all possible engagements between the BLUE and RED configurations instead of running the simulation one engagement at a time.

Since the focus of this effort is to gain insight into the relationship between the λ_{PFE} value and the effectiveness of a networked force, the agent-based simulation rules of engagement were quite simple. The battlespace (i.e., "world") within the model is deliberately featureless in order to focus on the configurations themselves. The agents are randomly distributed across the battlespace at the beginning of each engagement. Engagements continued until either all of the Sensors and Influencers of one force were annihilated, or both forces were incapable of continued combat (i.e., neither side contained a functioning combat cycle). A single run of the agent-based model will result in a BLUE win, a RED win, or an undecided result.

During each time tick of the simulation, the following procedures are executed: "establish-links," "sense," "track," "shoot," "kill," "move-Influencer," "move-Sensor," and "reset." The "establish-links" procedure establishes the links defined by the "directed-link-breed" keyword earlier in the code. It does so by breed, thereby ensuring each Sensor and Influencer is connected to only one Decider.

```
to establish-links
  ask BDecider1s [
      ask BSensor1s [create-detection-to myself [set
          color blue] ]
      ask BInfluencer1s [create-order-from myself [set
          color blue] ] ]
```

At this time, two of the four necessary links (types 2 and 6) of the IACM combat cycle have been established in the simulation. Link type 3 and one of the two interpretations of link type 13 are established in the "sense" procedure. In this procedure, every Decider asks its assigned Sensors (i.e., "in-link-neighbors") to identify all adversary Sensors and Influencers within its sensing range (i.e., "s-range"). Upon identification, the specific "sensed" variable of the targeted agent for that particular opposing Decider is set to a value of "1." The "s-range" parameter remains constant for all Sensors, either BLUE or RED, over time. The "sense" procedure depicted below is repeated for every BLUE and RED Decider breed:

```
to sense
  ask BDecider1s [
    ask in-link-neighbors [
      ask RInfluencer1s in-radius s-range [set
        sensedBD1 1]
      ask RSensor1s in-radius s-range [set sensedBD1
        1]
      ask RInfluencer2s in-radius s-range [set
        sensedBD1 1]
      ask RSensor2s in-radius s-range [set sensedBD1
        1]
      ask RInfluencer3s in-radius s-range [set
```

```
      sensedBD1 1]
    ask RSensor3s in-radius s-range [set sensedBD1
      1]
    ask RInfluencer4s in-radius s-range [set
      sensedBD1 1]
    ask RSensor4s in-radius s-range [set sensedBD1
      1]
    ask RInfluencer5s in-radius s-range [set
      sensedBD1 1]
    ask RSensor5s in-radius s-range [set sensedBD1
      1] ] ]
```

The remaining links necessary to complete the IACM combat cycles (link type 15 and the alternate interpretation of link type 13) are established by the "track," "shoot," and "kill" procedures. During the "track" procedure, every Decider asks its assigned Influencers (i.e., "out-link-neighbors") to identify all adversary Sensors and Influencers within its influencing range (i.e., "i-range"). Upon identification, the targeted agent is linked to that particular Influencer using the "create-LOF-from-myself" keyword. The "i-range" parameter remains constant for all Influencers, either BLUE or RED, over time. The "track" procedure depicted below is repeated for every BLUE and RED Decider breed:

```
to track
  ask BDecider1s [
    ask out-link-neighbors [
      ask RInfluencer1s in-radius i-range [create-LOF-
        from myself]
      ask RSensor1s in-radius i-range [create-LOF-from
        myself]
      ask RInfluencer2s in-radius i-range [create-LOF-
        from myself]
      ask RSensor2s in-radius i-range [create-LOF-from
        myself]
      ask RInfluencer3s in-radius i-range [create-LOF-
        from myself]
      ask RSensor3s in-radius i-range [create-LOF-from
        myself]
      ask RInfluencer4s in-radius i-range [create-LOF-
        from myself]
      ask RSensor4s in-radius i-range [create-LOF-from
        myself]
      ask RInfluencer4s in-radius i-range [create-LOF-
        from myself]
      ask RSensor4s in-radius i-range [create-LOF-from
```

```
myself] ] ]
```

Now that the complete IACM combat cycle has been established, the "shoot" and "kill" procedures represent its execution. During this procedure, each Decider directs its assigned Influencers to identify the single closest opposing Sensor or Influencer with which it shares a "LOF" link. This limits all Influencers to the same rate of fire of one targeted node per time tick. Identification is portrayed by setting the "dead" variable equal to "1."

```
to shoot
  ask BDecider1s [
    ask out-link-neighbors [
      ask out-link-neighbors [
        let $targets-sensed turtles with [(sensedBD1 =
          1) and (side = 2)]
          if any? $targets-sensed [
            ask min-one-of $targets-sensed [distance
            myself] [set dead 1] ] ] ] ]
```

Following this identification, the "kill" procedure deletes all agents that have been "sensed," "tracked" and "shot." The purpose of separating the "kill" procedure from the "shoot" procedure is to allow simultaneous shots, thereby precluding any advantage that would be gained by the order of execution of the "shoot" procedure code.

```
to kill
  ask turtles with [(dead = 1)] [die]
end
```

The collective effect of the "sense," "track," "shoot," and "kill" procedures is to require that a Sensor and an Influencer must be assigned to the same Decider and within their respective "s-range" and "i-range" in order to successfully complete a combat cycle (i.e., eliminate the targeted node).

Upon completion of all combat cycle execution, all remaining Sensors and Influencers are moved. The "move-Influencer" procedure directs all Influencers to move towards the nearest opposing Sensor or Influencer that has been "sensed" by a friendly Sensor assigned to the same Decider. If there are no qualifying opposing Sensors or influencers, then the Influencer will not move. Each time tick includes five iterative moves of a distance of "1" that are sequential between Deciders and sides in order to preclude any advantage of moving first or last. An example iteration for one Decider follows below:

```
to move-Influencer
  ask BDecider1s [
    ask out-link-neighbors [
      let $targets-sensed turtles with [(sensedBD1 =
      1) and (side = 2)]
        if any? $targets-sensed [
          set heading towards min-one-of $targets-
          sensed [distance myself] forward 1 ] ] ]
```

The "move-Sensor" procedure directs all Sensors to move towards the nearest Sensor or Influencer that is not currently "sensed" by a friendly Sensor assigned to the same Decider. This procedure is necessary to enable both sides to eventually target those opposing Sensors and Influencers that did not start the simulation within any friendly Sensor's "s-range."

```
to move-Sensor
  ask BDecider1s [
      ask in-link-neighbors [
        let $targets-sensed turtles with [(sensedBD1 =
        1) and (side = 2)]
          if not any? $targets-sensed [
            let $targets-unsensed turtles with
            [(sensedBD1 = 0) and (side = 2)]
              if any? $targets-unsensed [
                let $nearest-unsensed min-one-of
                $targets-unsensed [distance myself]
                set heading towards $nearest-unsensed
                forward 1 ] ] ] ]
```

The final procedure during each time tick is "reset." During this procedure, all "sensed" variables are reset to "0" and all links, to include the "LOF" tracking links, are deleted in preparation for the "establish-links," "sense," "track," "shoot," "kill," "move-Influencer," "move-Sensor," and "reset" procedures for the next time tick.

EXPERIMENTAL APPROACH

The scientific method was employed in this research in the following manner:

Step 1: Identify the Problem. The problem statement was derived from the background information provided in Chapter 1: How should an Information Age combat force be organized in order to optimize its effectiveness?

Step 2: Review Literature. Chapter 2 provided an in depth description of the problem statement and a discussion of the relevant literature to date.

Step 3: Formulate Hypothesis. Is the λ_{PFE} a significant indicator of combat effectiveness?

Step 4: Design Empirical Test of Hypothesis. Identify an X-Y-X-1 networked force with a number of meaningfully different combinations that is small enough to allow a comprehensive test of each configuration against each other configuration. Create an agent-based simulation model that will accurately model the IACM.

Step 5: Conduct Experiment. Conduct 30 agent-based simulation replications of each engagement (a test of any two configurations against each other).

Step 6: Analyze Data. Provided in Chapter 4.

Step 7: Draw Conclusions. Provided in Chapter 5.

CHAPTER IV

MODELING RESULTS

MODELING A 7-3-7-1 NETWORK

The initial experiment consisted of all possible force-on-force engagements of the 42 meaningfully different configurations of two networked forces (BLUE and RED) containing 7 Sensors, 3 Deciders, 7 Influencers, and 1 Target (i.e., two 7-3-7-1 forces). The sole Target node is representative of all the opposing nodes vulnerable to destruction. Additionally, the capabilities for each of these node types were identical between the forces. Since each of these configurations contains the same numbers of Sensors, Deciders, and Influencers, differing only in their connectivity, it is most likely that any difference in performance would be a consequence of this connectivity difference. A comprehensive test of each of these 42 configurations against each other required 1,764 different engagements. Each engagement was represented by 30 replications, each with a random distribution of the BLUE and RED nodes across the battlespace. Each replication resulted in a BLUE win, a RED win, or an undecided result (i.e., neither side contained a functioning combat cycle). The number of replications yielding an undecided result was 2,717 (5.13% of the 52,920 total). A graphical representation of the results is presented in Figure 7, with the shaded surface representing those engagements where the probability of a BLUE win was greater than 0.5 and the unshaded surface representing those engagements where the probability of a BLUE win was less than 0.5.

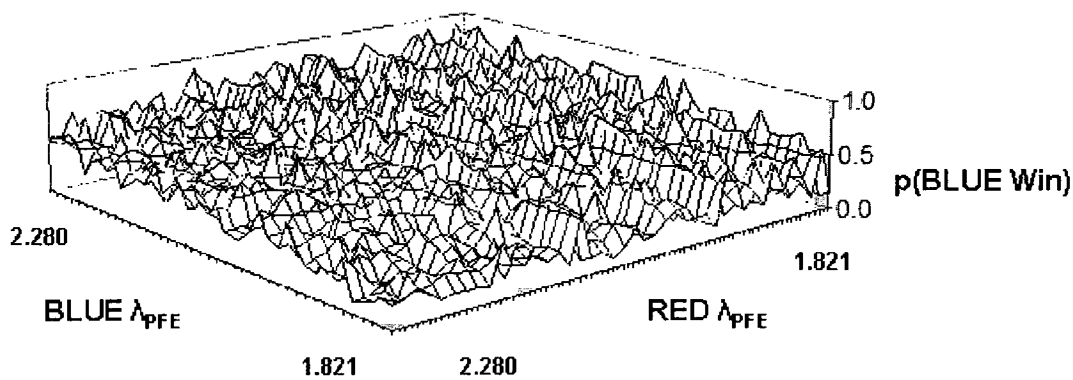

Figure 7. The probability of a BLUE Win for each of the 42 BLUE configurations (with λ_{BLUE} values varying from 1.821 to 2.280) against each of the 42 RED configurations (also with λ_{RED} values varying from 1.821 to 2.280). Surface values > 0.5 are shaded gray and surface values < 0.5 are unshaded.

These initial results indicate that as the BLUE force is organized to enhance its networked effects (i.e., the λ_{PFE} value increases), its effectiveness generally increases. While the resulting surface is far from smooth, a general trend does appear: the smaller the λ_{PFE} value, the smaller the probability of a win. This trend becomes more apparent in Figure 8, where the probability of a BLUE win for any particular configuration is averaged over all RED configurations. Note that many BLUE configurations had an identical λ_{PFE} value (there were 13 unique λ_{PFE} values for the 42 configurations). The average probability of a BLUE win for each of the configurations of the 7-3-7-1, 8-3-8-1 and 9-5-9-1 forces are listed in Tables 19, 20 and 21.

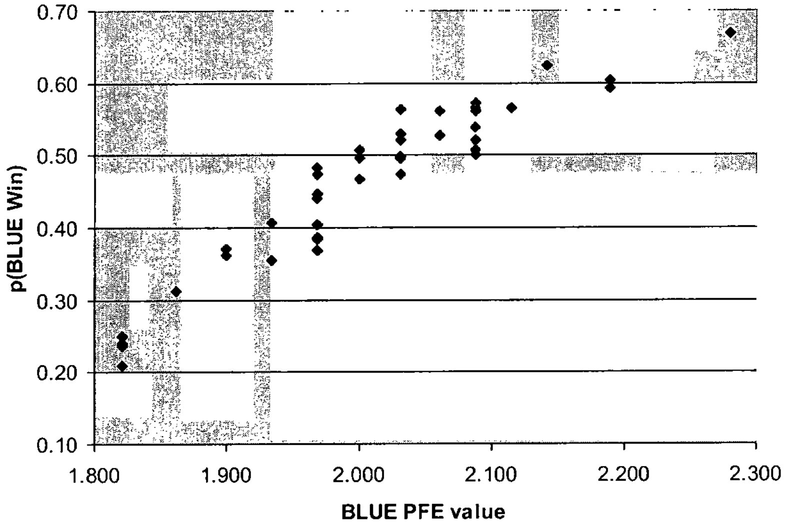

Figure 8. The average probability of a BLUE win by λ_{PFE} for 42 configurations of a 7-3-7-1 BLUE network.

Clearly, it appears that the probability of a BLUE win increases for those BLUE configurations with a greater λ_{PFE} value. A simple linear regression confirms this with a coefficient of determination (R^2) equal to 0.896 for the following equation (see Table 25 for the full regression results):

$$y = 1.0162(x) - 1.5780$$

where, y = the average probability of a BLUE win for that configuration

x = the λ_{PFE} value of a configuration

Utilizing an ordinal scale, these BLUE configurations can be ranked from 1 to 42 based on their average probability of a BLUE win. Doing so gives us some insight as to why there is a positive correlation between the λ_{PFE} value and the probability of a win (see Figure 9).

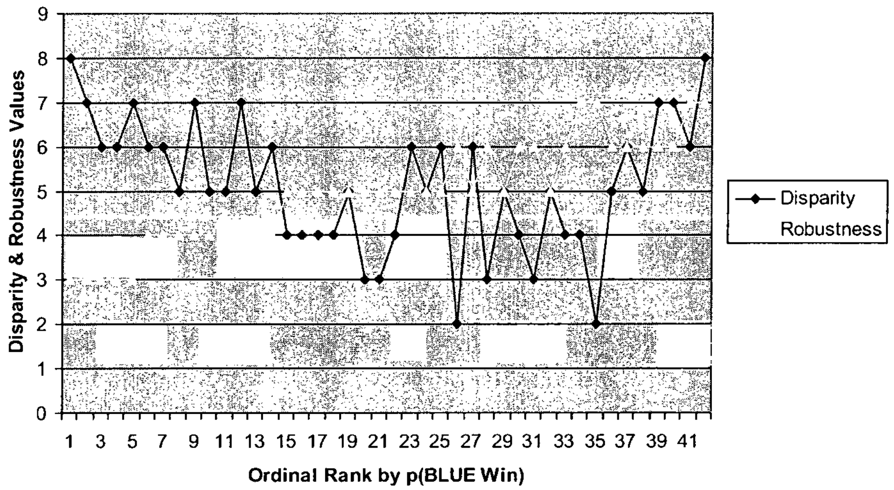

Figure 9. The disparity and robustness values of the 42 configurations of a 7-3-7-1 BLUE network.

Disparity is defined as the sum of the inequality of distribution of Sensors and Influencers across the Deciders. This can be mathematically expressed as:

$$Disparity = [\max(S_n)-\min(S_n)] + [\max(I_n)-\min(I_n)]$$

where, S_n = the number of Sensors assigned to each of n Deciders

I_n = the number of Influencers assigned to each of n Deciders

For example, Configuration #3 of a 7-3-7-1 networked force has linked five Sensors to Decider_1, one to Decider_2, and one to Decider_3, for a Sensor Disparity value of four. Configuration #3 has also linked one Influencer to Decider_1, four to Decider_2, and two to Decider_3, for an Influencer Disparity value of three. The total Disparity value of Configuration #3 is seven. The greater the disparity of a configuration, the greater the likelihood of an extreme high or low value for p(Win).

Whether a configuration's disparity has a positive or negative impact on the p(Win) is determined by the balance of Sensors and Influencers for each Decider within that configuration. $Decider_1$ of Configuration #3 has five Sensors but only one Influencer, while $Decider_2$ has one Sensor and four Influencers. This lack of balance has a negative impact on the configuration's performance as it reduces the minimum number of nodes that can be lost before a portion of the force is rendered combat ineffective. If the sole Influencer linked to $Decider_1$ is lost, then the five Sensors are combat ineffective as the information collected by the Sensors cannot be acted on. Consequently, the average probability of the BLUE Win for Configuration #3 was only 0.2365, which was the second-worst performance across all of the 7-3-7-1 configurations (see Table 19). Barabasi (2002) uses the term "robustness" to describe a network's resilience to failure due to the loss of some of its nodes.

Robustness is defined as the minimum number of nodes lost that would render the entire configuration unable to destroy any more enemy nodes. This can be mathematically expressed as:

$$Robustness = [\min(S_1, I_1)] + [\min(S_2, I_2)] + \cdots + [\min(S_n, I_n)]$$

where, S_n = the number of Sensors assigned to Decider n

I_n = the number of Influencers assigned to Decider n

In essence, the robustness value reflects the rate of the reduction of the λ_{PFE} value over time. The greater the robustness value, the longer a configuration will maintain combat effectiveness. Configurations that were more robust have a greater p(Win) value, while less robust configurations had a lower p(Win) value. Consequently, while disparity can be positive or negative in impact, robustness is predominantly positive.

MODELING AN 8-3-8-1 NETWORK

Adding just one additional Sensor and one additional Influencer to the 7-3-7-1 force increased the number of meaningful combinations to 78. A comprehensive test of each of these 78 configurations against each other required 6,084 different engagements. Each engagement was represented by 30 replications, each with a random distribution of the BLUE and RED nodes across the battlespace. Each replication resulted in a BLUE win, a RED win, or an undecided result (i.e., both BLUE and RED are unable to complete the annihilation of the opposing force due to a lack of Sensors or Influencers). The number of replications yielding an undecided result was 8,820 (4.83%). The average p(win) value for each of these configurations is shown in Figure 10. There were 24 unique λ_{PFE} values for the 78 configurations.

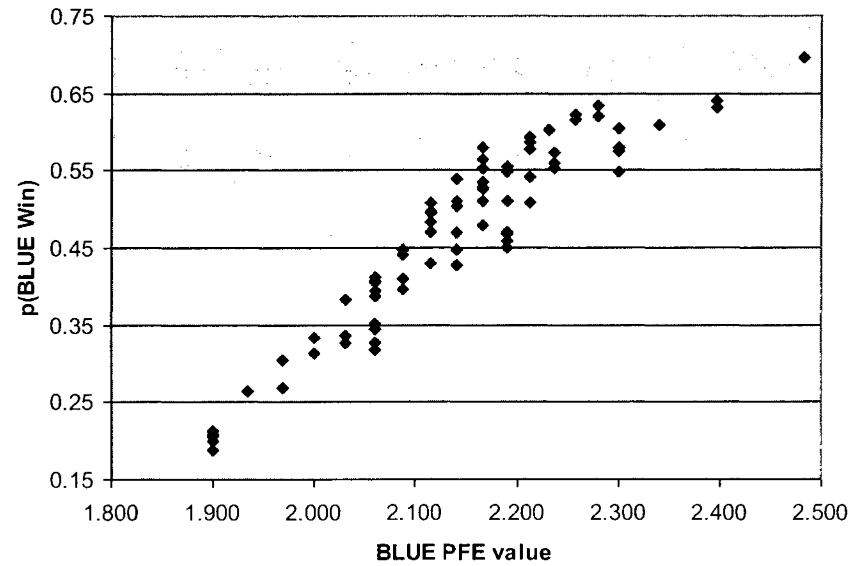

Figure 10. The average probability of a BLUE win by λ_{PFE} for 78 configurations of an 8-3-8-1 BLUE network.

These results also indicate that as the BLUE force is organized to enhance its networked effects (i.e., the λ_{PFE} value increases), its effectiveness generally increases. A simple linear regression confirms this with a coefficient of determination (R^2) equal to 0.876 for the following equation (see Table 26 for the full regression results):

$$y = 0.9484(x) - 1.5633$$

where, y = the average probability of a BLUE win for that configuration

x = the λ_{PFE} value of a configuration

Again, utilizing an ordinal scale these BLUE configurations can be ranked from 1 to 78 based on their average probability of a BLUE win (subsequently ordered by their λ_{PFE} values for those configurations with equal p(Win) values, where possible). The resulting trends in disparity and robustness are shown in Figure 11.

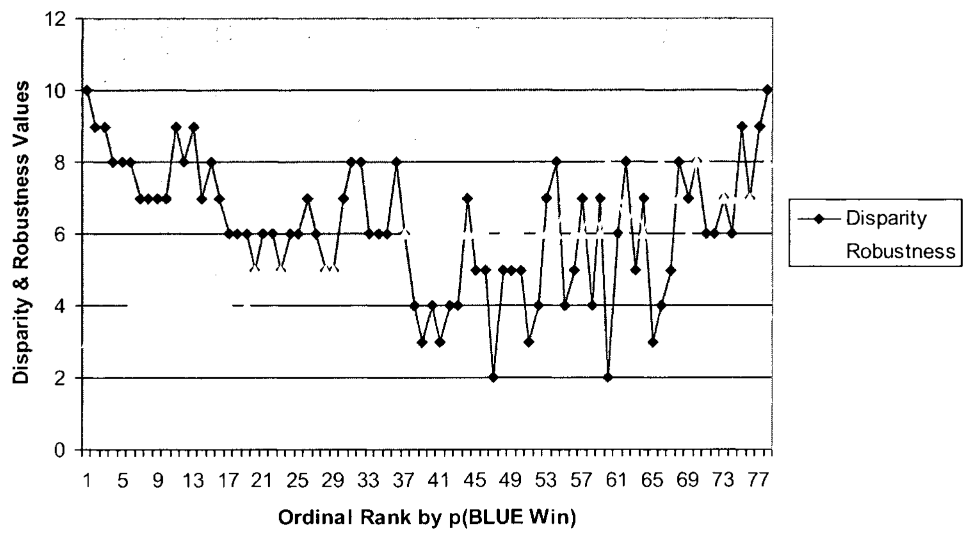

Figure 11. The disparity and robustness values of the 78 configurations of an 8-3-8-1 BLUE network.

MODELING A 9-5-9-1 NETWORK

Since increasing the number of Sensors and Influencers substantiated the initial results, the next logical step was to determine the impact of increasing the number of Deciders. A 9-5-9-1 force was selected since the number of meaningful combinations (95) was considerable, yet small enough to be modeled in a reasonable amount of time (it required approximately 78 hours of agent-based simulation model runtime). A comprehensive test of each of these 95 configurations against each other required 9,025 different engagements. Each engagement was represented by 30 replications, each with a random distribution of the BLUE and RED nodes across the battlespace. Each replication resulted in a BLUE win, a RED win, or an undecided result (i.e., both BLUE and RED are unable to complete the annihilation of the opposing force due to a lack of Sensors or Influencers). The number of replications yielding an undecided result was 19,892 (7.35%). The average p(win) value for each of these configurations is shown in Figure 12. There were 13 unique λ_{PFE} values for the 95 configurations.

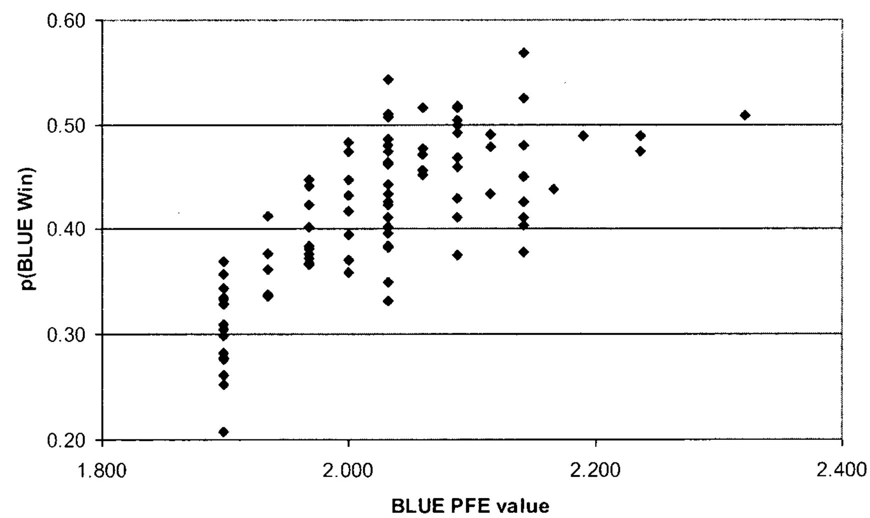

Figure 12. The average probability of a BLUE win by λ_{PFE} for 95 configurations of a 9-5-9-1 BLUE network.

Note that the highest p(win) value does not belong to the configuration with the highest λ_{PFE} value. This indicates that there is some other correlating factor, and the dramatic reduction in the coefficient of determination (R^2) to 0.519 for the resulting equation confirms that (see Table 27):

$$y = 0.5861(x) - 0.7736$$

where, y = the average probability of a BLUE win for that configuration
x = the λ_{PFE} value of a configuration

One cause of this was the marked increase in the number of ties. The first two experiments averaged 1.54 ties per engagement (for the 7-3-7-1 force), and 1.45 ties per engagement (for the 8-3-8-1 force). This experiment averaged 2.20 ties per engagement. Eliminating the tie results from the p(Win) calculations results in the correlation depicted in Figure 13.

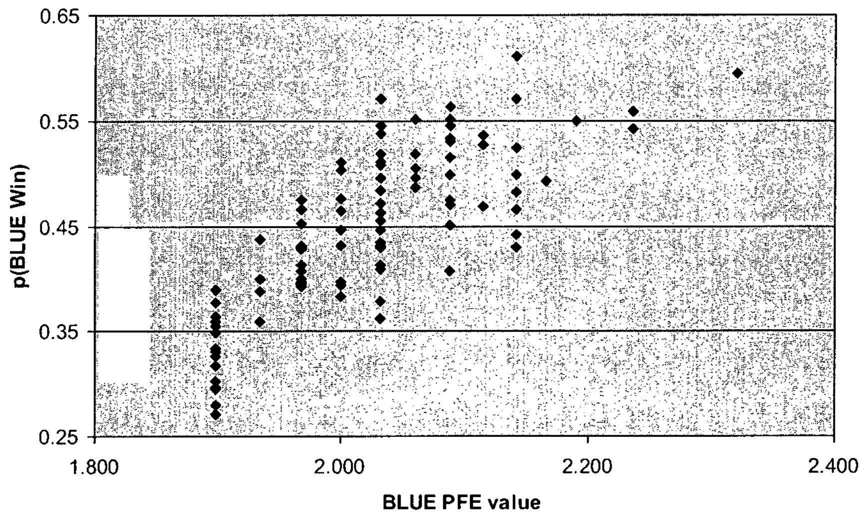

Figure 13. The average probability of a BLUE win by λ_{PFE} for 95 configurations of a 9-5-9-1 BLUE network (not including ties).

Note that the highest p(win) value still does not belong to the configuration with the highest λ_{PFE} value. The results of the linear regression (see Table 28) yield the following equation:

$$y = 0.7147(x) - 1.0002$$

where, y = the average probability of a BLUE win for that configuration
x = the λ_{PFE} value of a configuration

However, the coefficient of determination (R^2) only increased moderately to a value of 0.621. Another significant correlating factor remained.

An unexpected result of the increase of the number of assets to 9-5-9-1 force was the significant reduction in the number of unique λ_{PFE} values to 13, for a ratio of 13.68%. The first two experiments contained much larger ratios of unique λ_{PFE} values: 13 of 42 (30.95%) for the 7-3-7-1 force, and 24 of 78 (30.77%) for the 8-3-8-1 force. The impact of this reduction is a greater disparity of p(Win) values for any particular unique λ_{PFE} value. Consequently, the coefficient of determination (R^2) value is reduced significantly. This is the point at which the robustness value becomes particularly useful. Once again, utilizing an ordinal scale these BLUE configurations can be ranked from 1 to 95 based on their average probability of a BLUE win (subsequently ordered by their λ_{PFE} values for those configurations with equal p(Win) values, where possible). The resulting trends in disparity and robustness are shown in Figure 14.

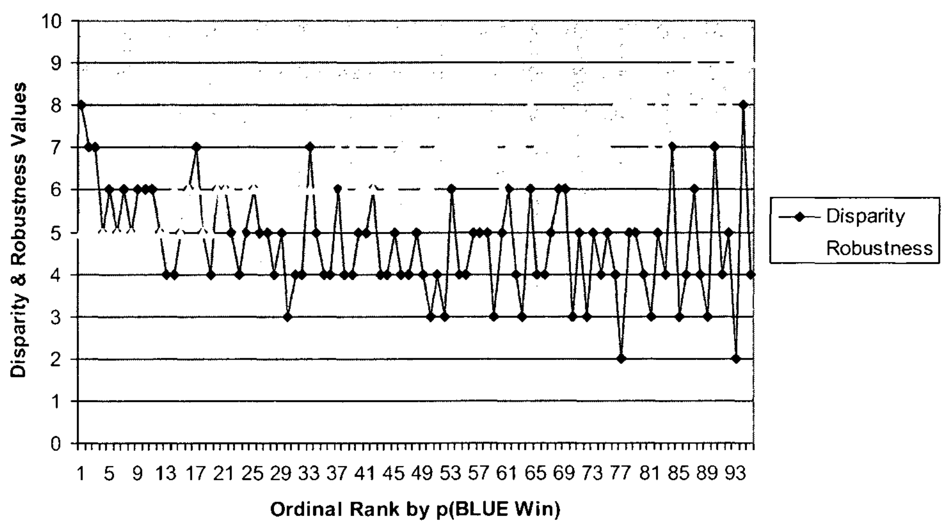

Figure 14. The disparity and robustness values of the 95 configurations of a 9-5-9-1 BLUE network.

Since the λ_{PFE} values now have a reduced correlation to the values of p(Win), the robustness value becomes much more useful in discriminating between configurations. For example, 20 of the 95 configurations share the λ_{PFE} value of 2.031 (see Table 24). Which configuration should have a greater average p(Win) value? By looking at the robustness value for each configuration (which vary from 6 to 9), we see that the sole configuration with a robustness value of 9 (Configuration #93) has the highest average p(Win) value. This average p(Win) value of 0.5425 is substantially larger than any configurations with an equal λ_{PFE} value. Configuration #93 has the highest average p(Win) among the configurations with equal λ_{PFE} values regardless of whether ties are included or not (0.5710 without including ties). Note that the configuration with the highest average p(Win) (Configuration #43) is not the configuration with the highest λ_{PFE} value (although its λ_{PFE} value of 2.141 is quite high), but one of the configurations with the highest robustness value. A regression analysis of both the λ_{PFE} value and the robustness value yields a significant increase in the coefficient of determination (R^2) from a value of 0.621 to 0.850 (see Table 29) and provides the following equation:

$$y = [0.0997(x_1) + 0.0613(x_2)] - 0.1617$$

where, y = the average probability of a BLUE win for that configuration
x_1 = the λ_{PFE} value of a configuration
x_2 = the robustness value of a configuration

The increase in the value of the coefficient of determination (R^2) remains significant even if we include the tie results again. In this case the value of R^2 is 0.805 (see Table 30) and provides the following equation:

$$y = [(-0.0307)(x_1) + 0.0615(x_2)] + 0.0678$$

where, y = the average probability of a BLUE win for that configuration
x_1 = the λ_{PFE} value of a configuration
x_2 = the robustness value of a configuration

CHAPTER V

CONCLUSION

ACHIEVING THE RESEARCH PURPOSE

The purpose of this research was to gain insight into how an Information Age combat force should be organized in order to optimize its effectiveness. Given the lack of quantifiable metrics that are able to discriminate between various networked forces that differ solely in their arrangement (see Table 31), this research represents an initial attempt to determine the utility of the Perron-Frobenius Eigenvalue (λ_{PFE}) as a measure of the ability of a network to produce feedback effects in general and combat power specifically in the case of the Information Age Combat Model (Cares 2005). This was accomplished by testing various force configurations of the Information Age Combat Model (IACM) in an agent-based simulation model.

The results of the agent-based simulation modeling presented in this work indicate that the value of the λ_{PFE} is a significant measurement of the performance of an Information Age combat force. A force organized for greater networked effects (i.e., the value of λ_{PFE} is greater) will defeat a force with equal assets and capabilities, but organized in a less-optimal manner, more often. The coefficient of determination (R^2) of both the 7-3-7-1 and 8-3-8-1 networked forces showed a strong degree of correlation between the λ_{PFE} value and the average probability of a Win.

While the λ_{PFE} value alone was a sufficient indicator for networked forces with three Deciders, it was not sufficient for a networked force with five. The greater number of configurations with the same λ_{PFE} value reduced the effectiveness of the λ_{PFE} value in discriminating between those configurations. Consequently, another metric was required. Disparity and robustness factors were introduced in this research to improve the effectiveness of the λ_{PFE} value as a quantifiable metric of network performance, and can be utilized in other similar research as quantifying factors. By utilizing both the λ_{PFE} value and the robustness value, the coefficient of determination (R^2) for the 9-5-9-1 networked force showed a strong degree of correlation with the average probability of a Win. No other quantifiable network metrics are able to consistently discriminate between configurations that differ by a single link, regardless of the significance of that link.

RECOMMENDATIONS FOR FUTURE WORK

The success of this research warrants further exploration in a number of areas:

- Determining the mathematical function to identify the meaningfully different configurations of any particular networked force. The 1,758 meaningfully different configurations considered in this research were identified through the manual application of a simple algorithm based on the numbers of unique values for the distributions of Sensors and Influencers across the Deciders. This process of identification would be simplified through the application of the actual mathematical function that defines the numbers of meaningfully different configurations.
- Expanding the experiment to include horizontal links (i.e., those links that connect nodes linked to different Deciders) within the tested configurations. How should these horizontal links be interpreted within the IACM? What effects will these horizontal links have on the λ_{PFE} value and the average probability of a win? Will the λ_{PFE} remain a significant measurement of the performance of an Information Age combat force?
- Investigating the effects of replacing the deterministic links with stochastic values between 0 and 1 representing the probabilities of detect, communicate and kill. What effects will these stochastic values have on the λ_{PFE} value and the average probability of a win? Will the λ_{PFE} remain a significant measurement of the performance of an Information Age combat force?
- Determining the value of a Sensor relative to an Influencer. What are the marginal values of Sensors and Influencers? Are those values equal?
- Expanding the research to larger networks (i.e., significantly increasing the numbers of Sensors, Deciders, and Influencers). Will the λ_{PFE} remain a significant measurement of the performance of an Information Age combat force?

- Investigating the effects of making the Deciders a vulnerable target for opposing Influencers. What effect will this vulnerability have on the average probability of a win? Will the λ_{PFE} remain a significant measurement of the performance of an Information Age combat force?
- Developing the simulation to include specific capabilities of individual Sensors, Deciders, and Influencers such as rates of movement, sensing and influencing ranges, search patterns, survivability, etc.
- Investigate the effects of engagements between forces without the same numbers of assets. Will a smaller, more-optimally organized force defeat a larger force? How many assets can optimization offset?

SUMMARY

Meeting the security challenges of the twenty-first century will require innovative approaches to implementing the functions of command and control in the Information Age. Selecting the right approach requires an understanding of the potential networked effects of a combat force resulting from quantifiable metrics that properly represent the interdependencies and complex local behaviors of Information Age warfare. The content of this section is mostly included in Deller, et al. (2009).

While the IACM can provide useful insights to inform the difficult decisions and trade-offs during the ongoing transformation into an Information Age combat force, it can also be generalized beyond attrition applications. Since the IACM is focused on network capability, the abstract representation of the acts of sensing (information), deciding, and influencing (whether it be combat or some other action taken) enable it to model almost any activity involving planning and decision making. The nodes represent capabilities and the connections the accessibility. Each mission consists of certain required capabilities and their connectivity can therefore be represented as a network. The likelihood of success for a mission can be directly mapped to the connectivity of its required capabilities. That connectivity can be informed by the quantitative metric (λ_{PFE} value) addressed in this work. As such, this research is not limited to merely addressing net-centric attrition but represents a quantitative approach towards analyzing all networked operations in a more rigorous manner and can be applied to the kinds of NCO as described in the NATO Code of Best Practice for C2 Assessment (2002), among others.

REFERENCES

Alberts, David S., Garstka, John J., and Stein, Frederick P. (1999). *Network Centric Warfare*, CCRP Press, Washington D.C.

Alberts, David S., Garstka, John J., Hayes, Richard E., and Signori, David A. (2001). *Understanding Information Age Warfare*, CCRP Press, Washington D.C.

Alberts, David S., and Hayes, Richard E. (2003). *Power to the Edge*, CCRP Press, Washington D.C.

Alberts, David S. (2007). Agility, Focus, and Convergence: The Future of Command and Control. *The International C2 Journal*, Vol. 1 No. 1, http://www.dodccrp.org/html4/journal_v1n1.html (accessed April 24, 2008).

Arquilla, John, and Ronfeldt, David (1993). Cyberwar is Coming! *Comparative Strategy* 12: 141-165.

Arquilla, John, and Ronfeldt, David (1996). *The Advent of Netwar*, RAND National Defense Research Institute, Santa Monica.

Arquilla, John, and Ronfeldt, David (1998). *The Zapatista Netwar in Mexico*, RAND National Defense Research Institute, Santa Monica.

Arquilla, John, and Ronfeldt, David (2000). *Swarming & The Future of Conflict*, RAND National Defense Research Institute, Santa Monica.

Arquilla, John, and Ronfeldt, David (2001). *Networks and Netwars*, RAND National Defense Research Institute, Santa Monica.

Barabasi, A (2002). *Linked*, Perseus Publishing, Cambridge.

Bell, Michael I. (2008). Private Communication.

Cares, Jeff (2005). *Distributed Networked Operations*. iUniverse, New York.

Cebrowski, Arthur K., and Garstka, John J. (1998). Network-Centric Warfare: Its Origin and Future. *United States Naval Institute Proceedings* 124: 28-35.

Chartrand, Gary (1984). *Introductory Graph Theory*. Dover Publications.

Deller, Sean, Bell, Michael I., Rabadi, Ghaith A., Bowling, Shannon R., and Tolk, Andreas (2009). Applying the Information Age Combat Model: Quantitative Analysis of Network Centric Operations. Accepted for publication in *The International C2 Journal*.

Department of Defense (2001). *Network Centric Warfare: Department of Defense Report to Congress.*

Epstein, Joshua M. (1985). *The Calculus of Conventional War: Dynamic Analysis Without Lanchester Theory*, The Brookings Institute, Washington, D.C.

Hammes, Thomas X. (1994). The Evolution of War: The Fourth Generation, *Marine Corps Gazette.*

Hammes, Thomas X. (2005). *The Sling and the Stone*, Zenith Press, St Paul.

Jain, Sanjay, and Krishna, Sandeep (1998). Autocatalytic Sets and the Growth of Complexity in an Evolutionary Model. *Physical Review Letters* 81: 5684-5687.

Jain, Sanjay, and Krishna, Sandeep (1999). Emergence and growth of complex networks in adaptive systems. *Computer Physics Communications* 121-122: 116-121.

Joint Publication 1-02 (2001). Department of Defense dictionary of military and associated terms. http://www.dtic.mil/doctrine/jel/new_pubs/jp1_02.pdf (accessed, September 19, 2008).

Kewley, Robert H., and Embrechts, Mark J. (2008). A Multiagent System for Tactical Control of Automated Forces *The International C2 Journal 2*(2), 1-28.

Lanchester, Frederick W. (1914). Aircraft in Warfare: The Dawn of the Fourth Arm, *Engineering* 48: 422-423.

Lind, William S., Nightengale, Keith, Schmitt, John F., Sutton, Joseph W., and Wilson, Gary I. (1989). The Changing Face of War: Into the Fourth Generation, *Marine Corps Gazette*, Oct: 22-26.

Lind, William S. (2005). Draft FMFM 1-A Fourth Generation War, viewed 21 September 2005, <http://www.d-n-i.net/lind/lind_7_06_05.htm>.

Ling, Michael F., Moon, Terry, and Kruzins, Ed. (2005). Proposed Network Centric Warfare Metrics: From Connectivity to the OODA Cycle, *Military Operations Research*, 10: 5-13.

NATO Code of Best Practice for C2 Assessment, Rev.Ed. CCRP Press, (2002).

Reid, Darryn J., Goodman, Graham, Johnson, Wayne, Giffin, Ralph E. (2007). All That Glisters: Is Network-Centric Warfare Really Scientific?, *Defense & Security Analysis*, 21: 335-367.

Sun Tzu. *The Art of War*, Ed. James Clavell, Delacorte Press, New York, (1983).

Toffler, Alvin, and Toffler, Heidi. (1993). *War and Anti-War: Survival at the Dawn of the Twenty-First Century*, Little, Brown and Company, New York.

Van Creveld, Martin. (2000). Through A Glass Darkly: Some Reflections on the Future of War, *Naval War College Review*, Fall 2000.

Wilenski, Uri (1999). *NetLogo.* Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. http://ccl.northwestern.edu/netlogo.

APPENDIX A

MEANINGFUL CONFIGURATIONS

Table 5. All Possible Configurations of a 4-3-4-1 Network.

Configuration Number	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3
1	2	1	1	2	1	1
2	2	1	1	1	2	1
3	2	1	1	1	1	2
4	1	2	1	2	1	1
5	1	2	1	1	2	1
6	1	2	1	1	1	2
7	1	1	2	2	1	1
8	1	1	2	1	2	1
9	1	1	2	1	1	2

Table 6. All Possible Configurations of a 5-3-5-1 Network.

Number	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Influencers linked to Decider1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3
1	3	1	1	3	1	1
2	3	1	1	1	3	1
3	3	1	1	1	1	3
4	1	3	1	3	1	1
5	1	3	1	1	3	1
6	1	3	1	1	1	3
7	1	1	3	3	1	1
8	1	1	3	1	3	1
9	1	1	3	1	1	3
10	3	1	1	2	2	1
11	3	1	1	1	2	2
12	3	1	1	2	1	2
13	1	3	1	2	2	1
14	1	3	1	1	2	2
15	1	3	1	2	1	2
16	1	1	3	2	2	1
17	1	1	3	1	2	2
18	1	1	3	2	1	2
19	2	2	1	3	1	1
20	2	2	1	1	3	1
21	2	2	1	1	1	3
22	2	1	2	3	1	1
23	2	1	2	1	3	1
24	2	1	2	1	1	3
25	1	2	2	3	1	1
26	1	2	2	1	3	1
27	1	2	2	1	1	3
28	2	2	1	2	2	1
29	2	2	1	2	1	2
30	2	2	1	1	2	2
31	2	1	2	2	2	1
32	2	1	2	2	1	2
33	2	1	2	1	2	2
34	1	2	2	2	2	1
35	1	2	2	2	1	2
36	1	2	2	1	2	2

Table 7. All Possible Configurations of a 7-3-7-1 Network.

Number	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3
1	5	1	1	5	1	1
2	5	1	1	1	5	1
3	5	1	1	1	1	5
4	1	5	1	5	1	1
5	1	5	1	1	5	1
6	1	5	1	1	1	5
7	1	1	5	5	1	1
8	1	1	5	1	5	1
9	1	1	5	1	1	5
10	5	1	1	4	2	1
11	5	1	1	4	1	2
12	5	1	1	2	4	1
13	5	1	1	2	1	4
14	5	1	1	1	4	2
15	5	1	1	1	2	4
16	1	5	1	4	2	1
17	1	5	1	4	1	2
18	1	5	1	2	4	1
19	1	5	1	2	1	4
20	1	5	1	1	4	2
21	1	5	1	1	2	4
22	1	1	5	4	2	1
23	1	1	5	4	1	2
24	1	1	5	2	4	1
25	1	1	5	2	1	4
26	1	1	5	1	4	2
27	1	1	5	1	2	4
28	5	1	1	3	3	1
29	5	1	1	3	1	3
30	5	1	1	1	3	3
31	1	5	1	3	3	1
32	1	5	1	3	1	3
33	1	5	1	1	3	3
34	1	1	5	3	3	1
35	1	1	5	3	1	3
36	1	1	5	1	3	3
37	5	1	1	3	2	2
38	5	1	1	2	3	2
39	5	1	1	2	2	3
40	1	5	1	3	2	2
41	1	5	1	2	3	2
42	1	5	1	2	2	3
43	1	1	5	3	2	2
44	1	1	5	2	3	2
45	1	1	5	2	2	3
46	4	2	1	5	1	1
47	4	2	1	1	5	1
48	4	2	1	1	1	5
49	4	1	2	5	1	1
50	4	1	2	1	5	1
51	4	1	2	1	1	5
52	2	4	1	5	1	1
53	2	4	1	1	5	1
54	2	4	1	1	1	5
55	2	1	4	5	1	1
56	2	1	4	1	5	1
57	2	1	4	1	1	5
58	1	4	2	5	1	1
59	1	4	2	1	5	1
60	1	4	2	1	1	5
61	1	2	4	5	1	1
62	1	2	4	1	5	1
63	1	2	4	1	1	5
64	4	2	1	4	2	1
65	4	2	1	4	1	2
66	4	2	1	2	4	1
67	4	2	1	2	1	4
68	4	2	1	1	4	2
69	4	2	1	1	2	4
70	4	1	2	4	2	1
71	4	1	2	4	1	2
72	4	1	2	2	4	1
73	4	1	2	2	1	4
74	4	1	2	1	4	2
75	4	1	2	1	2	4
76	2	4	1	4	2	1
77	2	4	1	4	1	2
78	2	4	1	2	4	1
79	2	4	1	2	1	4
80	2	4	1	1	4	2

Table 7 (continued).

Number	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3
81	2	4	1	1	2	4
82	2	1	4	4	2	1
83	2	1	4	4	1	2
84	2	1	4	2	4	1
85	2	1	4	2	1	4
86	2	1	4	1	4	2
87	2	1	4	1	2	4
88	1	4	2	4	2	1
89	1	4	2	4	1	2
90	1	4	2	2	4	1
91	1	4	2	2	1	4
92	1	4	2	1	4	2
93	1	4	2	1	2	4
94	1	2	4	4	2	1
95	1	2	4	4	1	2
96	1	2	4	2	4	1
97	1	2	4	2	1	4
98	1	2	4	1	4	2
99	1	2	4	1	2	4
100	4	2	1	3	3	1
101	4	2	1	3	1	3
102	4	2	1	1	3	3
103	4	1	2	3	3	1
104	4	1	2	3	1	3
105	4	1	2	1	3	3
106	2	4	1	3	3	1
107	2	4	1	3	1	3
108	2	4	1	1	3	3
109	2	1	4	3	3	1
110	2	1	4	3	1	3
111	2	1	4	1	3	3
112	1	4	2	3	3	1
113	1	4	2	3	1	3
114	1	4	2	1	3	3
115	1	2	4	3	3	1
116	1	2	4	3	1	3
117	1	2	4	1	3	3
118	4	2	1	3	2	2
119	4	2	1	2	3	2
120	4	2	1	2	2	3

Number	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3
121	4	1	2	3	2	2
122	4	1	2	2	3	2
123	4	1	2	2	2	3
124	2	4	1	3	2	2
125	2	4	1	2	3	2
126	2	4	1	2	2	3
127	2	1	4	3	2	2
128	2	1	4	2	3	2
129	2	1	4	2	2	3
130	1	4	2	3	2	2
131	1	4	2	2	3	2
132	1	4	2	2	2	3
133	1	2	4	3	2	2
134	1	2	4	2	3	2
135	1	2	4	2	2	3
136	3	3	1	5	1	1
137	3	3	1	1	5	1
138	3	3	1	1	1	5
139	3	1	3	5	1	1
140	3	1	3	1	5	1
141	3	1	3	1	1	5
142	1	3	3	5	1	1
143	1	3	3	1	5	1
144	1	3	3	1	1	5
145	3	3	1	4	2	1
146	3	3	1	4	1	2
147	3	3	1	2	4	1
148	3	3	1	2	1	4
149	3	3	1	1	4	2
150	3	3	1	1	2	4
151	3	1	3	4	2	1
152	3	1	3	4	1	2
153	3	1	3	2	4	1
154	3	1	3	2	1	4
155	3	1	3	1	4	2
156	3	1	3	1	2	4
157	1	3	3	4	2	1
158	1	3	3	4	1	2
159	1	3	3	2	4	1
160	1	3	3	2	1	4

Table 7 (continued).

Number	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3
161	1	3	3	1	4	2
162	1	3	3	1	2	4
163	3	3	1	3	3	1
164	3	3	1	3	1	3
165	3	3	1	1	3	3
166	3	1	3	3	3	1
167	3	1	3	3	1	3
168	3	1	3	1	3	3
169	1	3	3	3	3	1
170	1	3	3	3	1	3
171	1	3	3	1	3	3
172	3	3	1	3	2	2
173	3	3	1	2	3	2
174	3	3	1	2	2	3
175	3	1	3	3	2	2
176	3	1	3	2	3	2
177	3	1	3	2	2	3
178	1	3	3	3	2	2
179	1	3	3	2	3	2
180	1	3	3	2	2	3
181	3	2	2	5	1	1
182	3	2	2	1	5	1
183	3	2	2	1	1	5
184	2	3	2	5	1	1
185	2	3	2	1	5	1
186	2	3	2	1	1	5
187	2	2	3	5	1	1
188	2	2	3	1	5	1
189	2	2	3	1	1	5
190	3	2	2	4	2	1
191	3	2	2	4	1	2
192	3	2	2	2	4	1
193	3	2	2	2	1	4
194	3	2	2	1	4	2
195	3	2	2	1	2	4
196	2	3	2	4	2	1
197	2	3	2	4	1	2
198	2	3	2	2	4	1
199	2	3	2	2	1	4
200	2	3	2	1	4	2

Number	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3
201	2	3	2	1	2	4
202	2	2	3	4	2	1
203	2	2	3	4	1	2
204	2	2	3	2	4	1
205	2	2	3	2	1	4
206	2	2	3	1	4	2
207	2	2	3	1	2	4
208	3	2	2	3	3	1
209	3	2	2	3	1	3
210	3	2	2	1	3	3
211	2	3	2	3	3	1
212	2	3	2	3	1	3
213	2	3	2	1	3	3
214	2	2	3	3	3	1
215	2	2	3	3	1	3
216	2	2	3	1	3	3
217	3	2	2	3	2	2
218	3	2	2	2	3	2
219	3	2	2	2	2	3
220	2	3	2	3	2	2
221	2	3	2	2	3	2
222	2	3	2	2	2	3
223	2	2	3	3	2	2
224	2	2	3	2	3	2
225	2	2	3	2	2	3

Table 8. Meaningful Combinations of an X-3-X-1 Network.

	Sensor Allocations for Y = 3			Influencer Allocations for Y = 3			Number of Meaningful Configurations
X = 3	1	1	1	1	1	1	1
						Total =	1
X = 4	2	1	1	2	1	1	2
						Total =	2
X = 5	3	1	1	3	1	1	2
	3	1	1	2	2	1	2
	2	2	1	3	1	1	2
	2	2	1	2	2	1	2
						Total =	8
X = 6	4	1	1	4	1	1	2
	4	1	1	3	2	1	3
	4	1	1	2	2	2	1
	3	2	1	4	1	1	3
	3	2	1	3	2	1	6
	3	2	1	2	2	2	1
	2	2	2	4	1	1	1
	2	2	2	3	2	1	1
	2	2	2	2	2	2	1
						Total =	19
X = 7	5	1	1	5	1	1	2
	5	1	1	4	2	1	3
	5	1	1	3	3	1	2
	5	1	1	3	2	2	2
	4	2	1	5	1	1	3
	4	2	1	4	2	1	6
	4	2	1	3	3	1	3
	4	2	1	3	2	2	3
	3	3	1	5	1	1	2
	3	3	1	4	2	1	3
	3	3	1	3	3	1	2
	3	3	1	3	2	2	2
	3	2	2	5	1	1	2
	3	2	2	4	2	1	3
	3	2	2	3	3	1	2
	3	2	2	3	2	2	2
						Total =	42
X = 8	6	1	1	6	1	1	2
	6	1	1	5	2	1	3
	6	1	1	4	3	1	3

Table 8 (continued).

	Sensor Allocations for Y = 3			Influencer Allocations for Y = 3			Number of Meaningful Configurations
X = 8 (cont.)	6	1	1	4	2	2	2
	6	1	1	3	3	2	2
	5	2	1	6	1	1	3
	5	2	1	5	2	1	6
	5	2	1	4	3	1	6
	5	2	1	4	2	2	3
	5	2	1	3	3	2	3
	4	3	1	6	1	1	3
	4	3	1	5	2	1	6
	4	3	1	4	3	1	6
	4	3	1	4	2	2	3
	4	3	1	3	3	2	3
	4	2	2	6	1	1	2
	4	2	2	5	2	1	3
	4	2	2	4	3	1	3
	4	2	2	4	2	2	2
	4	2	2	3	3	2	2
	3	3	2	6	1	1	2
	3	3	2	5	2	1	3
	3	3	2	4	3	1	3
	3	3	2	4	2	2	2
	3	3	2	3	3	2	2
						Total =	66
X = 9	7	1	1	7	1	1	2
	7	1	1	6	2	1	3
	7	1	1	5	3	1	3
	7	1	1	5	2	2	2
	7	1	1	4	4	1	2
	7	1	1	4	3	2	3
	7	1	1	3	3	3	1
	6	2	1	7	1	1	3
	6	2	1	6	2	1	6
	6	2	1	5	3	1	6
	6	2	1	5	2	2	3
	6	2	1	4	4	1	3
	6	2	1	4	3	2	6
	6	2	1	3	3	3	1
	5	3	1	7	1	1	3
	5	3	1	6	2	1	6

Table 8 (continued).

	Sensor Allocations for Y = 3			Influencer Allocations for Y = 3			Number of Meaningful Configurations
X = 9 (cont.)	5	3	1	5	3	1	6
	5	3	1	5	2	2	3
	5	3	1	4	4	1	3
	5	3	1	4	3	2	6
	5	3	1	3	3	3	1
	5	2	2	7	1	1	2
	5	2	2	6	2	1	3
	5	2	2	5	3	1	3
	5	2	2	5	2	2	2
	5	2	2	4	4	1	2
	5	2	2	4	3	2	3
	5	2	2	3	3	3	1
	4	4	1	7	1	1	2
	4	4	1	6	2	1	3
	4	4	1	5	3	1	3
	4	4	1	5	2	2	2
	4	4	1	4	4	1	2
	4	4	1	4	3	2	3
	4	4	1	3	3	3	1
	4	3	2	7	1	1	3
	4	3	2	6	2	1	6
	4	3	2	5	3	1	6
	4	3	2	5	2	2	3
	4	3	2	4	4	1	3
	4	3	2	4	3	2	6
	4	3	2	3	3	3	1
	3	3	3	7	1	1	1
	3	3	3	6	2	1	1
	3	3	3	5	3	1	1
	3	3	3	5	2	2	1
	3	3	3	4	4	1	1
	3	3	3	4	3	2	1
	3	3	3	3	3	3	1
						Total =	68
X = 10	8	1	1	8	1	1	2
	8	1	1	7	2	1	3
	8	1	1	6	3	1	3
	8	1	1	6	2	2	2
	8	1	1	5	4	1	3

Table 8 (continued).

	Sensor Allocations for Y = 3			Influencer Allocations for Y = 3			Number of Meaningful Configurations
X=10 (cont.)	8	1	1	5	3	2	3
	8	1	1	4	4	2	2
	8	1	1	4	3	3	2
	7	2	1	8	1	1	3
	7	2	1	7	2	1	6
	7	2	1	6	3	1	6
	7	2	1	6	2	2	3
	7	2	1	5	4	1	6
	7	2	1	5	3	2	6
	7	2	1	4	4	2	3
	7	2	1	4	3	3	3
	6	3	1	8	1	1	3
	6	3	1	7	2	1	6
	6	3	1	6	3	1	6
	6	3	1	6	2	2	3
	6	3	1	5	4	1	6
	6	3	1	5	3	2	6
	6	3	1	4	4	2	3
	6	3	1	4	3	3	3
	6	2	2	8	1	1	2
	6	2	2	7	2	1	3
	6	2	2	6	3	1	3
	6	2	2	6	2	2	2
	6	2	2	5	4	1	3
	6	2	2	5	3	2	3
	6	2	2	4	4	2	2
	6	2	2	4	3	3	2
	5	4	1	8	1	1	3
	5	4	1	7	2	1	6
	5	4	1	6	3	1	6
	5	4	1	6	2	2	3
	5	4	1	5	4	1	6
	5	4	1	5	3	2	6
	5	4	1	4	4	2	3
	5	4	1	4	3	3	3
	5	3	2	8	1	1	3
	5	3	2	7	2	1	6
	5	3	2	6	3	1	6
	5	3	2	6	2	2	3

Table 8 (continued).

	Sensor Allocations for Y = 3			Influencer Allocations for Y = 3			Number of Meaningful Configurations
X=10 (cont.)	5	3	2	5	4	1	6
	5	3	2	5	3	2	6
	5	3	2	4	4	2	3
	5	3	2	4	3	3	3
	4	4	2	8	1	1	2
	4	4	2	7	2	1	3
	4	4	2	6	3	1	3
	4	4	2	6	2	2	2
	4	4	2	5	4	1	3
	4	4	2	5	3	2	3
	4	4	2	4	4	2	2
	4	4	2	4	3	3	2
	4	3	3	8	1	1	2
	4	3	3	7	2	1	3
	4	3	3	6	3	1	3
	4	3	3	6	2	2	2
	4	3	3	5	4	1	3
	4	3	3	5	3	2	3
	4	3	3	4	4	2	2
	4	3	3	4	3	3	2
						Total =	224

Table 9. Meaningful Combinations of an X-4-X-1 Network.

	Sensor Allocations for Y = 4				Influencer Allocations for Y = 4				Number of Meaningful Configurations
X = 4	1	1	1	1	1	1	1	1	1
								Total =	1
X = 5	2	1	1	1	2	1	1	1	2
								Total =	2
X = 6	3	1	1	1	3	1	1	1	2
	3	1	1	1	2	2	1	1	2
	2	2	1	1	3	1	1	1	2
	2	2	1	1	2	2	1	1	3
								Total =	9
X = 7	4	1	1	1	4	1	1	1	2
	4	1	1	1	3	2	1	1	3
	4	1	1	1	2	2	2	1	2
	3	2	1	1	4	1	1	1	3
	3	2	1	1	3	2	1	1	7
	3	2	1	1	2	2	2	1	3
	2	2	2	1	4	1	1	1	2
	2	2	2	1	3	2	1	1	3
	2	2	2	1	2	2	2	1	2
								Total =	27
X = 8	5	1	1	1	5	1	1	1	2
	5	1	1	1	4	2	1	1	3
	5	1	1	1	3	3	1	1	2
	5	1	1	1	3	2	2	1	3
	5	1	1	1	2	2	2	2	1
	4	2	1	1	5	1	1	1	3
	4	2	1	1	4	2	1	1	7
	4	2	1	1	3	3	1	1	4
	4	2	1	1	3	2	2	1	7
	4	2	1	1	2	2	2	2	1
	3	3	1	1	5	1	1	1	2
	3	3	1	1	4	2	1	1	4
	3	3	1	1	3	3	1	1	3
	3	3	1	1	3	2	2	1	4
	3	3	1	1	2	2	2	2	1
	3	2	2	1	5	1	1	1	3
	3	2	2	1	4	2	1	1	7
	3	2	2	1	3	3	1	1	4
	3	2	2	1	3	2	2	1	7
	3	2	2	1	2	2	2	2	1

Table 9 (continued).

	Sensor Allocations for Y = 4				Influencer Allocations for Y = 4				Number of Meaningful Configurations
X = 8 (cont.)	2	2	2	2	5	1	1	1	1
	2	2	2	2	4	2	1	1	1
	2	2	2	2	3	3	1	1	1
	2	2	2	2	3	2	2	1	1
	2	2	2	2	2	2	2	2	1
								Total =	69
X = 9	6	1	1	1	6	1	1	1	2
	6	1	1	1	5	2	1	1	3
	6	1	1	1	4	3	1	1	3
	6	1	1	1	4	2	2	1	3
	6	1	1	1	3	3	2	1	3
	6	1	1	1	3	2	2	2	2
	5	2	1	1	6	1	1	1	3
	5	2	1	1	5	2	1	1	7
	5	2	1	1	4	3	1	1	7
	5	2	1	1	4	2	2	1	7
	5	2	1	1	3	3	2	1	7
	5	2	1	1	3	2	2	2	3
	4	3	1	1	6	1	1	1	3
	4	3	1	1	5	2	1	1	7
	4	3	1	1	4	3	1	1	7
	4	3	1	1	4	2	2	1	7
	4	3	1	1	3	3	2	1	7
	4	3	1	1	3	2	2	2	3
	4	2	2	1	6	1	1	1	3
	4	2	2	1	5	2	1	1	7
	4	2	2	1	4	3	1	1	7
	4	2	2	1	4	2	2	1	7
	4	2	2	1	3	3	2	1	7
	4	2	2	1	3	2	2	2	3
	3	3	2	1	6	1	1	1	3
	3	3	2	1	5	2	1	1	7
	3	3	2	1	4	3	1	1	7
	3	3	2	1	4	2	2	1	7
	3	3	2	1	3	3	2	1	7
	3	3	2	1	3	2	2	2	3
	3	2	2	2	6	1	1	1	2
	3	2	2	2	5	2	1	1	3
	3	2	2	2	4	3	1	1	3

Table 9 (continued).

	Sensor Allocations for Y = 4				Influencer Allocations for Y = 4				Number of Meaningful Configurations
X = 9 (cont.)	3	2	2	2	4	2	2	1	3
	3	2	2	2	3	3	2	1	3
	3	2	2	2	3	2	2	2	2
								Total =	160
X = 10	7	1	1	1	7	1	1	1	2
	7	1	1	1	6	2	1	1	3
	7	1	1	1	5	3	1	1	3
	7	1	1	1	5	2	2	1	3
	7	1	1	1	4	4	1	1	2
	7	1	1	1	4	3	2	1	4
	7	1	1	1	4	2	2	2	2
	7	1	1	1	3	3	3	1	2
	7	1	1	1	3	3	2	2	2
	6	2	1	1	7	1	1	1	3
	6	2	1	1	6	2	1	1	7
	6	2	1	1	5	3	1	1	7
	6	2	1	1	5	2	2	1	7
	6	2	1	1	4	4	1	1	4
	6	2	1	1	4	3	2	1	12
	6	2	1	1	4	2	2	2	3
	6	2	1	1	3	3	3	1	3
	6	2	1	1	3	3	2	2	4
	5	3	1	1	7	1	1	1	3
	5	3	1	1	6	2	1	1	7
	5	3	1	1	5	3	1	1	7
	5	3	1	1	5	2	2	1	7
	5	3	1	1	4	4	1	1	4
	5	3	1	1	4	3	2	1	12
	5	3	1	1	4	2	2	2	3
	5	3	1	1	3	3	3	1	3
	5	3	1	1	3	3	2	2	4
	5	2	2	1	7	1	1	1	3
	5	2	2	1	6	2	1	1	7
	5	2	2	1	5	3	1	1	7
	5	2	2	1	5	2	2	1	7
	5	2	2	1	4	4	1	1	4
	5	2	2	1	4	3	2	1	12
	5	2	2	1	4	2	2	2	3
	5	2	2	1	3	3	3	1	3

Table 9 (continued).

	Sensor Allocations for Y = 4				Influencer Allocations for Y = 4				Number of Meaningful Configurations
X = 10 (cont.)	5	2	2	1	3	3	2	2	4
	4	4	1	1	7	1	1	1	2
	4	4	1	1	6	2	1	1	4
	4	4	1	1	5	3	1	1	4
	4	4	1	1	5	2	2	1	4
	4	4	1	1	4	4	1	1	3
	4	4	1	1	4	3	2	1	6
	4	4	1	1	4	2	2	2	2
	4	4	1	1	3	3	3	1	2
	4	4	1	1	3	3	2	2	3
	4	3	2	1	7	1	1	1	4
	4	3	2	1	6	2	1	1	12
	4	3	2	1	5	3	1	1	12
	4	3	2	1	5	2	2	1	12
	4	3	2	1	4	4	1	1	6
	4	3	2	1	4	3	2	1	24
	4	3	2	1	4	2	2	2	4
	4	3	2	1	3	3	3	1	4
	4	3	2	1	3	3	2	2	6
	4	2	2	2	7	1	1	1	2
	4	2	2	2	6	2	1	1	3
	4	2	2	2	5	3	1	1	3
	4	2	2	2	5	2	2	1	3
	4	2	2	2	4	4	1	1	2
	4	2	2	2	4	3	2	1	4
	4	2	2	2	4	2	2	2	2
	4	2	2	2	3	3	3	1	2
	4	2	2	2	3	3	2	2	2
	3	3	3	1	7	1	1	1	2
	3	3	3	1	6	2	1	1	3
	3	3	3	1	5	3	1	1	3
	3	3	3	1	5	2	2	1	3
	3	3	3	1	4	4	1	1	2
	3	3	3	1	4	3	2	1	4
	3	3	3	1	4	2	2	2	2
	3	3	3	1	3	3	3	1	2
	3	3	3	1	3	3	2	2	2
	3	3	2	2	7	1	1	1	2
	3	3	2	2	6	2	1	1	4

Table 9 (continued).

	Sensor Allocations for Y = 4				Influencer Allocations for Y = 4				Number of Meaningful Configurations
X = 10 (cont.)	3	3	2	2	5	3	1	1	4
	3	3	2	2	5	2	2	1	4
	3	3	2	2	4	4	1	1	3
	3	3	2	2	4	3	2	1	6
	3	3	2	2	4	2	2	2	2
	3	3	2	2	3	3	3	1	2
	3	3	2	2	3	3	2	2	3
								Total =	363

Table 10. Meaningful Combinations of an X-5-X-1 Network.

	Sensor Allocations for Y = 5					Influencer Allocations for Y = 5					Number of Meaningful Configurations
X = 5	1	1	1	1	1	1	1	1	1	1	1
										Total =	1
X = 6	2	1	1	1	1	2	1	1	1	1	2
										Total =	2
X = 7	3	1	1	1	1	3	1	1	1	1	2
	3	1	1	1	1	2	2	1	1	1	2
	2	2	1	1	1	3	1	1	1	1	2
	2	2	1	1	1	2	2	1	1	1	3
										Total =	9
X = 8	4	1	1	1	1	4	1	1	1	1	2
	4	1	1	1	1	3	2	1	1	1	3
	4	1	1	1	1	2	2	2	1	1	2
	3	2	1	1	1	4	1	1	1	1	3
	3	2	1	1	1	3	2	1	1	1	7
	3	2	1	1	1	2	2	2	1	1	4
	2	2	2	1	1	4	1	1	1	1	2
	2	2	2	1	1	3	2	1	1	1	4
	2	2	2	1	1	2	2	2	1	1	3
										Total =	30
X = 9	5	1	1	1	1	5	1	1	1	1	2
	5	1	1	1	1	4	2	1	1	1	3
	5	1	1	1	1	3	3	1	1	1	2
	5	1	1	1	1	3	2	2	1	1	3
	5	1	1	1	1	2	2	2	2	1	2
	4	2	1	1	1	5	1	1	1	1	3
	4	2	1	1	1	4	2	1	1	1	7
	4	2	1	1	1	3	3	1	1	1	4
	4	2	1	1	1	3	2	2	1	1	8
	4	2	1	1	1	2	2	2	2	1	3
	3	3	1	1	1	5	1	1	1	1	2
	3	3	1	1	1	4	2	1	1	1	4
	3	3	1	1	1	3	3	1	1	1	3
	3	3	1	1	1	3	2	2	1	1	5
	3	3	1	1	1	2	2	2	2	1	2
	3	2	2	1	1	5	1	1	1	1	3
	3	2	2	1	1	4	2	1	1	1	8
	3	2	2	1	1	3	3	1	1	1	5
	3	2	2	1	1	3	2	2	1	1	11
	3	2	2	1	1	2	2	2	2	1	3
	2	2	2	2	1	5	1	1	1	1	2
	2	2	2	2	1	4	2	1	1	1	3
	2	2	2	2	1	3	3	1	1	1	2
	2	2	2	2	1	3	2	2	1	1	3
	2	2	2	2	1	2	2	2	2	1	2
										Total =	95

Table 10 (continued).

	Sensor Allocations for Y = 5					Influencer Allocations for Y = 5					Number of Meaningful Configurations
X = 10	6	1	1	1	1	6	1	1	1	1	2
	6	1	1	1	1	5	2	1	1	1	3
	6	1	1	1	1	4	3	1	1	1	3
	6	1	1	1	1	4	2	2	1	1	3
	6	1	1	1	1	3	3	2	1	1	3
	6	1	1	1	1	3	2	2	2	1	3
	6	1	1	1	1	2	2	2	2	2	1
	5	2	1	1	1	6	1	1	1	1	3
	5	2	1	1	1	5	2	1	1	1	7
	5	2	1	1	1	4	3	1	1	1	7
	5	2	1	1	1	4	2	2	1	1	8
	5	2	1	1	1	3	3	2	1	1	8
	5	2	1	1	1	3	2	2	2	1	7
	5	2	1	1	1	2	2	2	2	2	1
	4	3	1	1	1	6	1	1	1	1	3
	4	3	1	1	1	5	2	1	1	1	7
	4	3	1	1	1	4	3	1	1	1	7
	4	3	1	1	1	4	2	2	1	1	8
	4	3	1	1	1	3	3	2	1	1	8
	4	3	1	1	1	3	2	2	2	1	7
	4	3	1	1	1	2	2	2	2	2	1
	4	2	2	1	1	6	1	1	1	1	3
	4	2	2	1	1	5	2	1	1	1	8
	4	2	2	1	1	4	3	1	1	1	8
	4	2	2	1	1	4	2	2	1	1	11
	4	2	2	1	1	3	3	2	1	1	11
	4	2	2	1	1	3	2	2	2	1	8
	4	2	2	1	1	2	2	2	2	2	1
	3	3	2	1	1	6	1	1	1	1	3
	3	3	2	1	1	5	2	1	1	1	8
	3	3	2	1	1	4	3	1	1	1	8
	3	3	2	1	1	4	2	2	1	1	11
	3	3	2	1	1	3	3	2	1	1	11
	3	3	2	1	1	3	2	2	2	1	8
	3	3	2	1	1	2	2	2	2	2	1
	3	2	2	2	1	6	1	1	1	1	3
	3	2	2	2	1	5	2	1	1	1	7
	3	2	2	2	1	4	3	1	1	1	7
	3	2	2	2	1	4	2	2	1	1	8
	3	2	2	2	1	3	3	2	1	1	8
	3	2	2	2	1	3	2	2	2	1	7
	3	2	2	2	1	2	2	2	2	2	1
	2	2	2	2	2	6	1	1	1	1	1
	2	2	2	2	2	5	2	1	1	1	1
	2	2	2	2	2	4	3	1	1	1	1
	2	2	2	2	2	4	2	2	1	1	1
	2	2	2	2	2	3	3	2	1	1	1
	2	2	2	2	2	3	2	2	2	1	1
	2	2	2	2	2	2	2	2	2	2	1
										Total =	**248**

Table 11. Meaningful Combinations of an X-6-X-1 Network.

	Sensor Allocations for Y = 6						Influencer Allocations for Y = 6						Number of Meaningful Configurations
X = 6	1	1	1	1	1	1	1	1	1	1	1	1	1
												Total =	1
X = 7	2	1	1	1	1	1	2	1	1	1	1	1	2
												Total =	2
X = 8	3	1	1	1	1	1	3	1	1	1	1	1	2
	3	1	1	1	1	1	2	2	1	1	1	1	2
	2	2	1	1	1	1	3	1	1	1	1	1	2
	2	2	1	1	1	1	2	2	1	1	1	1	3
												Total =	9
X = 9	4	1	1	1	1	1	4	1	1	1	1	1	2
	4	1	1	1	1	1	3	2	1	1	1	1	3
	4	1	1	1	1	1	2	2	2	1	1	1	2
	3	2	1	1	1	1	4	1	1	1	1	1	3
	3	2	1	1	1	1	3	2	1	1	1	1	7
	3	2	1	1	1	1	2	2	2	1	1	1	4
	2	2	2	1	1	1	4	1	1	1	1	1	2
	2	2	2	1	1	1	3	2	1	1	1	1	4
	2	2	2	1	1	1	2	2	2	1	1	1	4
												Total =	31
X = 10	5	1	1	1	1	1	5	1	1	1	1	1	2
	5	1	1	1	1	1	4	2	1	1	1	1	3
	5	1	1	1	1	1	3	3	1	1	1	1	2
	5	1	1	1	1	1	3	2	2	1	1	1	3
	5	1	1	1	1	1	2	2	2	2	1	1	2
	4	2	1	1	1	1	5	1	1	1	1	1	3
	4	2	1	1	1	1	4	2	1	1	1	1	7
	4	2	1	1	1	1	3	3	1	1	1	1	4
	4	2	1	1	1	1	3	2	2	1	1	1	8
	4	2	1	1	1	1	2	2	2	2	1	1	4
	3	3	1	1	1	1	5	1	1	1	1	1	2
	3	3	1	1	1	1	4	2	1	1	1	1	4
	3	3	1	1	1	1	3	3	1	1	1	1	3
	3	3	1	1	1	1	3	2	2	1	1	1	5
	3	3	1	1	1	1	2	2	2	2	1	1	3
	3	2	2	1	1	1	5	1	1	1	1	1	3
	3	2	2	1	1	1	4	2	1	1	1	1	8
	3	2	2	1	1	1	3	3	1	1	1	1	5
	3	2	2	1	1	1	3	2	2	1	1	1	12
	3	2	2	1	1	1	2	2	2	2	1	1	5
	2	2	2	2	1	1	5	1	1	1	1	1	2
	2	2	2	2	1	1	4	2	1	1	1	1	4
	2	2	2	2	1	1	3	3	1	1	1	1	3
	2	2	2	2	1	1	3	2	2	1	1	1	5
	2	2	2	2	1	1	2	2	2	2	1	1	3
												Total =	105

Table 12. Meaningful Combinations of an X-7-X-1 Network.

	Sensor Allocations for Y = 7							Influencer Allocations for Y = 7							Number of Meaningful Configurations
X = 7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
														Total =	1
X = 8	2	1	1	1	1	1	1	2	1	1	1	1	1	1	2
														Total =	2
X = 9	3	1	1	1	1	1	1	3	1	1	1	1	1	1	2
	3	1	1	1	1	1	1	2	2	1	1	1	1	1	2
	2	2	1	1	1	1	1	3	1	1	1	1	1	1	2
	2	2	1	1	1	1	1	2	2	1	1	1	1	1	3
														Total =	9
X = 10	4	1	1	1	1	1	1	4	1	1	1	1	1	1	2
	4	1	1	1	1	1	1	3	2	1	1	1	1	1	3
	4	1	1	1	1	1	1	2	2	2	1	1	1	1	2
	3	2	1	1	1	1	1	4	1	1	1	1	1	1	3
	3	2	1	1	1	1	1	3	2	1	1	1	1	1	7
	3	2	1	1	1	1	1	2	2	2	1	1	1	1	4
	2	2	2	1	1	1	1	4	1	1	1	1	1	1	2
	2	2	2	1	1	1	1	3	2	1	1	1	1	1	4
	2	2	2	1	1	1	1	2	2	2	1	1	1	1	4
														Total =	31

APPENDIX B

PERRON-FROBENIUS EIGENVALUE VALUES

Table 13. λ_{PFE} Values for all Meaningful Configurations of a 7-3-7-1 Network.

Configuration Identification #	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3	λ_{PFE} value
0	5	1	1	5	1	1	2.280
1	5	1	1	1	5	1	1.821
2	5	1	1	4	2	1	2.190
3	5	1	1	1	4	2	1.821
4	5	1	1	2	1	4	1.968
5	5	1	1	3	3	1	2.088
6	5	1	1	1	3	3	1.821
7	5	1	1	3	2	2	2.088
8	5	1	1	2	3	2	1.968
9	4	2	1	5	1	1	2.190
10	4	2	1	1	5	1	1.968
11	4	2	1	1	1	5	1.821
12	4	2	1	4	2	1	2.141
13	4	2	1	4	1	2	2.115
14	4	2	1	2	4	1	2.031
15	4	2	1	2	1	4	1.934
16	4	2	1	1	4	2	1.934
17	4	2	1	1	2	4	1.861
18	4	2	1	3	3	1	2.088
19	4	2	1	3	1	3	2.031
20	4	2	1	1	3	3	1.899
21	4	2	1	3	2	2	2.060
22	4	2	1	2	3	2	2.000
23	4	2	1	2	2	3	1.968
24	3	1	3	5	1	1	2.088
25	3	1	3	1	5	1	1.821
26	3	1	3	4	2	1	2.031
27	3	1	3	1	4	2	1.899
28	3	1	3	2	1	4	2.088
29	3	1	3	3	3	1	1.968
30	3	1	3	3	1	3	2.088
31	3	1	3	3	2	2	2.031
32	3	1	3	2	3	2	1.968
33	3	2	2	5	1	1	2.088
34	3	2	2	1	5	1	1.968
35	3	2	2	4	2	1	2.060
36	3	2	2	1	4	2	1.968
37	3	2	2	2	1	4	2.000
38	3	2	2	3	3	1	2.031
39	3	2	2	1	3	3	1.968
40	3	2	2	3	2	2	2.031
41	3	2	2	2	3	2	2.000

Table 14. λ_{PFE} Values for all Meaningful Configurations of a 8-3-8-1 Network.

Configuration identification #	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3	λ_{PFE} value
0	6	1	1	6	1	1	2.483
1	6	1	1	1	6	1	1.899
2	6	1	1	5	2	1	2.397
3	6	1	1	1	5	2	1.899
4	6	1	1	2	1	5	2.060
5	6	1	1	4	3	1	2.300
6	6	1	1	1	4	3	1.899
7	6	1	1	3	1	4	2.190
8	6	1	1	4	2	2	2.300
9	6	1	1	2	4	2	2.060
10	6	1	1	3	3	2	2.190
11	6	1	1	2	3	3	2.060
12	5	2	1	6	1	1	2.397
13	5	2	1	1	6	1	2.060
14	5	2	1	1	1	6	1.899
15	5	2	1	5	2	1	2.340
16	5	2	1	5	1	2	2.231
17	5	2	1	2	5	1	2.141
18	5	2	1	2	1	5	2.031
19	5	2	1	1	5	2	2.031
20	5	2	1	1	2	5	1.934
21	5	2	1	4	3	1	2.280
22	5	2	1	4	1	3	2.236
23	5	2	1	3	4	1	2.213
24	5	2	1	3	1	4	2.141
25	5	2	1	1	4	3	2.000
26	5	2	1	1	3	4	1.968
27	5	2	1	4	2	2	2.258
28	5	2	1	2	4	2	2.115
29	5	2	1	2	2	4	2.060
30	5	2	1	3	3	2	2.190
31	5	2	1	3	2	3	2.166
32	5	2	1	2	3	3	2.088
33	4	3	1	6	1	1	2.300
34	4	3	1	1	6	1	2.190
35	4	3	1	1	1	6	1.899
36	4	3	1	5	2	1	2.280
37	4	3	1	5	1	2	2.236
38	4	3	1	2	5	1	2.213

Table 14 (continued).

Configuration Identification #	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3	A_{PFE} value
39	4	3	1	2	1	5	2.000
40	4	3	1	1	5	2	2.141
41	4	3	1	1	2	5	1.968
42	4	3	1	4	3	1	2.258
43	4	3	1	4	1	3	2.166
44	4	3	1	3	4	1	2.236
45	4	3	1	3	1	4	2.088
46	4	3	1	1	4	3	2.088
47	4	3	1	1	3	4	2.031
48	4	3	1	4	2	2	2.213
49	4	3	1	2	4	2	2.166
50	4	3	1	2	2	4	2.060
51	4	3	1	3	3	2	2.190
52	4	3	1	3	2	3	2.141
53	4	3	1	2	3	3	2.115
54	4	2	2	6	1	1	2.300
55	4	2	2	1	6	1	2.060
56	4	2	2	5	2	1	2.258
57	4	2	2	1	5	2	2.060
58	4	2	2	2	1	5	2.115
59	4	2	2	4	3	1	2.213
60	4	2	2	1	4	3	2.060
61	4	2	2	3	1	4	2.166
62	4	2	2	4	2	2	2.213
63	4	2	2	2	4	2	2.115
64	4	2	2	3	3	2	2.166
65	4	2	2	2	3	3	2.115
66	3	2	3	6	1	1	2.190
67	3	2	3	1	6	1	2.060
68	3	2	3	5	2	1	2.166
69	3	2	3	1	5	2	2.088
70	3	2	3	2	1	5	2.190
71	3	2	3	4	3	1	2.141
72	3	2	3	1	4	3	2.115
73	3	2	3	3	1	4	2.190
74	3	2	3	4	2	2	2.166
75	3	2	3	2	4	2	2.115
76	3	2	3	3	3	2	2.141
77	3	2	3	3	2	3	2.166

Table 15. λ_{PFE} Values for all Meaningful Configurations of a 9-5-9-1 Network.

Configuration Identification #	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Sensors linked to Decider 4	Number of Sensors linked to Decider 5	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3	Number of Influencers linked to Decider 4	Number of Influencers linked to Decider 5	λ_{PFE} Value
0	5	1	1	1	1	5	1	1	1	1	2.321
1	5	1	1	1	1	1	5	1	1	1	1.899
2	5	1	1	1	1	4	2	1	1	1	2.236
3	5	1	1	1	1	1	4	2	1	1	1.899
4	5	1	1	1	1	2	1	4	1	1	2.031
5	5	1	1	1	1	3	3	1	1	1	2.141
6	5	1	1	1	1	1	3	3	1	1	1.899
7	5	1	1	1	1	3	2	2	1	1	2.141
8	5	1	1	1	1	1	3	2	2	1	1.899
9	5	1	1	1	1	2	3	2	1	1	2.031
10	5	1	1	1	1	2	2	2	2	1	2.031
11	5	1	1	1	1	1	2	2	2	2	1.899
12	4	2	1	1	1	5	1	1	1	1	2.236
13	4	2	1	1	1	1	5	1	1	1	2.031
14	4	2	1	1	1	1	1	5	1	1	1.899
15	4	2	1	1	1	4	2	1	1	1	2.190
16	4	2	1	1	1	4	1	2	1	1	2.166
17	4	2	1	1	1	2	4	1	1	1	2.088
18	4	2	1	1	1	2	1	4	1	1	2.000
19	4	2	1	1	1	1	4	2	1	1	2.000
20	4	2	1	1	1	1	2	4	1	1	1.934
21	4	2	1	1	1	1	1	4	2	1	1.899
22	4	2	1	1	1	3	3	1	1	1	2.141
23	4	2	1	1	1	3	1	3	1	1	2.088
24	4	2	1	1	1	1	3	3	1	1	1.968
25	4	2	1	1	1	1	1	3	3	1	1.899
26	4	2	1	1	1	3	2	2	1	1	2.115
27	4	2	1	1	1	3	1	2	2	1	2.088
28	4	2	1	1	1	2	3	2	1	1	2.060
29	4	2	1	1	1	1	3	2	2	1	1.968
30	4	2	1	1	1	2	2	3	1	1	2.031
31	4	2	1	1	1	2	1	3	2	1	2.000

Table 15 (continued).

Configuration Identification #	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Sensors linked to Decider 4	Number of Sensors linked to Decider 5	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3	Number of Influencers linked to Decider 4	Number of Influencers linked to Decider 5	Λ_{PFE} value
32	4	2	1	1	1	1	2	3	2	1	**1.934**
33	4	2	1	1	1	1	1	3	2	2	**1.899**
34	4	2	1	1	1	2	2	2	2	1	**2.031**
35	4	2	1	1	1	2	1	2	2	2	**2.000**
36	4	2	1	1	1	1	2	2	2	2	**1.934**
37	3	3	1	1	1	5	1	1	1	1	**2.141**
38	3	3	1	1	1	1	1	5	1	1	**1.899**
39	3	3	1	1	1	4	2	1	1	1	**2.141**
40	3	3	1	1	1	4	1	2	1	1	**2.088**
41	3	3	1	1	1	2	1	4	1	1	**1.968**
42	3	3	1	1	1	1	1	4	2	1	**1.899**
43	3	3	1	1	1	3	3	1	1	1	**2.141**
44	3	3	1	1	1	3	1	3	1	1	**2.031**
45	3	3	1	1	1	1	1	3	3	1	**1.899**
46	3	3	1	1	1	3	2	2	1	1	**2.088**
47	3	3	1	1	1	3	1	2	2	1	**2.031**
48	3	3	1	1	1	2	2	3	1	1	**2.031**
49	3	3	1	1	1	2	1	3	2	1	**1.968**
50	3	3	1	1	1	1	1	3	2	2	**1.899**
51	3	3	1	1	1	2	2	2	2	1	**2.031**
52	3	3	1	1	1	2	1	2	2	2	**1.968**
53	3	2	2	1	1	5	1	1	1	1	**2.141**
54	3	2	2	1	1	1	5	1	1	1	**2.031**
55	3	2	2	1	1	1	1	1	5	1	**1.899**
56	3	2	2	1	1	4	2	1	1	1	**2.115**
57	3	2	2	1	1	4	1	1	2	1	**2.088**
58	3	2	2	1	1	2	4	1	1	1	**2.060**
59	3	2	2	1	1	2	1	1	4	1	**1.968**
60	3	2	2	1	1	1	4	2	1	1	**2.031**
61	3	2	2	1	1	1	4	1	2	1	**2.000**
62	3	2	2	1	1	1	2	1	4	1	**1.934**
63	3	2	2	1	1	1	1	1	4	2	**1.899**

Table 15 (continued).

Configuration identification #	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Sensors linked to Decider 4	Number of Sensors linked to Decider 5	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3	Number of Influencers linked to Decider 4	Number of Influencers linked to Decider 5	λ_{PFE} Value
64	3	2	2	1	1	3	3	1	1	1	2.088
65	3	2	2	1	1	3	1	1	3	1	2.031
66	3	2	2	1	1	1	3	3	1	1	2.031
67	3	2	2	1	1	1	3	1	3	1	1.968
68	3	2	2	1	1	1	1	1	3	3	1.899
69	3	2	2	1	1	3	2	2	1	1	2.088
70	3	2	2	1	1	3	2	1	2	1	2.060
71	3	2	2	1	1	3	1	1	2	2	2.031
72	3	2	2	1	1	2	3	2	1	1	2.060
73	3	2	2	1	1	2	3	1	2	1	2.031
74	3	2	2	1	1	2	2	1	3	1	2.000
75	3	2	2	1	1	2	1	1	3	2	1.968
76	3	2	2	1	1	1	3	2	2	1	2.000
77	3	2	2	1	1	1	3	1	2	2	1.968
78	3	2	2	1	1	1	2	2	3	1	1.968
79	3	2	2	1	1	1	2	1	3	2	1.934
80	3	2	2	1	1	2	2	2	2	1	2.031
81	3	2	2	1	1	2	2	1	2	2	2.000
82	3	2	2	1	1	1	2	2	2	2	1.968
83	2	2	2	2	1	5	1	1	1	1	2.141
84	2	2	2	2	1	1	1	1	1	5	1.899
85	2	2	2	2	1	4	2	1	1	1	2.115
86	2	2	2	2	1	4	1	1	1	2	2.088
87	2	2	2	2	1	2	1	1	1	4	1.968
88	2	2	2	2	1	3	3	1	1	1	2.088
89	2	2	2	2	1	3	1	1	1	3	2.031
90	2	2	2	2	1	3	2	2	1	1	2.088
91	2	2	2	2	1	3	2	1	1	2	2.060
92	2	2	2	2	1	2	2	1	1	3	2.000
93	2	2	2	2	1	2	2	2	2	1	2.031
94	2	2	2	2	1	2	2	2	1	2	2.031

APPENDIX C

UNIQUE PERRON-FROBENIUS EIGENVALUE VALUES

Table 16. Unique λ_{PFE} Values for all Meaningful Configurations of a 7-3-7-1 Network.

Count	Unique λ_{PFE} values	Number of Each Value	7-3-7-1 Configuration Identification #s
1	1.821	5	**1 3 6 11 25**
2	1.861	1	**17**
3	1.899	2	**20 27**
4	1.934	2	**15 16**
5	1.968	9	**4 8 10 23 29 32 34 36 39**
6	2.000	3	**22 37 41**
7	2.031	6	**14 19 26 31 38 40**
8	2.060	2	**21 35**
9	2.088	7	**5 7 18 24 28 30 33**
10	2.115	1	**13**
11	2.141	1	**12**
12	2.190	2	**2 9**
13	2.280	1	**0**
		42	**= Total Number of Configurations**

Table 17. Unique λ_{PFE} Values for all Meaningful Configurations of an 8-3-8-1 Network.

Count	Unique λ_{PFE} values	Number of Each Value	8-3-8-1 Configuration Identification #s
1	1.899	5	1 3 6 14 35
2	1.934	1	20
3	1.968	2	26 41
4	2.000	2	25 39
5	2.031	3	18 19 47
6	2.060	10	4 9 11 13 29 50 55 57 60 67
7	2.088	4	32 45 46 69
8	2.115	7	28 53 58 63 65 72 75
9	2.141	6	17 24 40 52 71 76
10	2.166	8	31 43 49 61 64 68 74 77
11	2.190	8	7 10 30 34 51 66 70 73
12	2.213	5	23 38 48 59 62
13	2.231	1	16
14	2.236	3	22 37 44
15	2.258	3	27 42 56
16	2.280	2	21 36
17	2.300	4	5 8 33 54
18	2.340	1	15
19	2.397	2	2 12
20	2.483	1	0
		78	= Total Number of Configurations

Table 18. Unique λ_{PFE} Values for all Meaningful Configurations of a 9-5-9-1 Network.

Count	Unique λ_{PFE} values	Number of Each Value	9-5-9-1 Configuration Identification #s
1	**1.899**	17	**1 3 6 8 11 14 21 25 33 38 42 45 50 55 63 68 84**
2	**1.934**	5	**20 32 36 62 79**
3	**1.968**	12	**24 29 41 49 52 59 67 75 77 78 82 87**
4	**2.000**	9	**18 19 31 35 61 74 76 81 92**
5	**2.031**	20	**4 9 10 13 30 34 44 47 48 51 54 60 65 66 71 73 80 89 93 94**
6	**2.060**	5	**28 58 70 72 91**
7	**2.088**	11	**17 23 27 40 46 57 64 69 86 88 90**
8	**2.115**	3	**26 56 85**
9	**2.141**	8	**5 7 22 37 39 43 53 83**
10	**2.166**	1	**16**
11	**2.190**	1	**15**
12	**2.236**	2	**2 12**
13	**2.321**	1	**0**
		95	**= Total Number of Configurations**

APPENDIX D

MODELING RESULTS

Table 19. Modeling Results for a 7-3-7-1 Network.

Configuration identification #	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3	λ_{PFE} value	Average p(BLUE Win) over all RED configurations	Ordinal Rank by p(BLUE Win)	Disparity of Sensors	Disparity of Influencers	Total Disparity	Robustness
1	5	1	1	1	5	1	1.821	0.2095	1	4	4	8	3
3	5	1	1	1	4	2	1.821	0.2365	2	4	3	7	3
6	5	1	1	1	3	3	1.821	0.2381	3	4	2	6	3
25	3	1	3	1	5	1	1.821	0.2405	4	2	4	6	3
11	4	2	1	1	1	5	1.821	0.2492	5	3	4	7	3
17	4	2	1	1	2	4	1.861	0.3127	6	3	3	6	4
16	4	2	1	1	4	2	1.934	0.3548	7	3	3	6	4
20	4	2	1	1	3	3	1.899	0.3619	8	3	2	5	4
4	5	1	1	2	1	4	1.968	0.3683	9	4	3	7	4
27	3	1	3	1	4	2	1.899	0.3714	10	2	3	5	4
8	5	1	1	2	3	2	1.968	0.3833	11	4	1	5	4
10	4	2	1	1	5	1	1.968	0.3857	12	3	4	7	4
34	3	2	2	1	5	1	1.968	0.4056	13	1	4	5	4
15	4	2	1	2	1	4	1.934	0.4063	14	3	3	6	4
29	3	1	3	3	3	1	1.968	0.4413	15	2	2	4	5
23	4	2	1	2	2	3	1.968	0.4468	16	3	1	4	5
36	3	2	2	1	4	2	1.968	0.4476	17	1	3	4	5
22	4	2	1	2	3	2	2.000	0.4667	18	3	1	4	5
19	4	2	1	3	1	3	2.031	0.4730	19	3	2	5	5
32	3	1	3	2	3	2	1.968	0.4738	20	2	1	3	5
39	3	2	2	1	3	3	1.968	0.4817	21	1	2	3	5
37	3	2	2	2	1	4	2.000	0.4952	22	1	3	4	5
14	4	2	1	2	4	1	2.031	0.4960	23	3	3	6	5
26	3	1	3	4	2	1	2.031	0.4984	24	2	3	5	5
24	3	1	3	5	1	1	2.088	0.5000	25	2	4	6	5
41	3	2	2	2	3	2	2.000	0.5063	26	1	1	2	6
5	5	1	1	3	3	1	2.088	0.5079	27	4	2	6	5
31	3	1	3	3	2	2	2.031	0.5198	28	2	1	3	6
7	5	1	1	3	2	2	2.088	0.5214	29	4	1	5	5
21	4	2	1	3	2	2	2.060	0.5270	30	3	1	4	6
38	3	2	2	3	3	1	2.031	0.5310	31	1	2	3	6
33	3	2	2	5	1	1	2.088	0.5397	32	1	4	5	5
35	3	2	2	4	2	1	2.060	0.5611	33	1	3	4	6
30	3	1	3	3	1	3	2.088	0.5619	34	2	2	4	7
40	3	2	2	3	2	2	2.031	0.5643	35	1	1	2	7
18	4	2	1	3	3	1	2.088	0.5651	36	3	2	5	6
13	4	2	1	4	1	2	2.115	0.5659	37	3	3	6	6
28	3	1	3	2	1	4	2.088	0.5714	38	2	3	5	6
9	4	2	1	5	1	1	2.190	0.5921	39	3	4	7	6
2	5	1	1	4	2	1	2.190	0.6040	40	4	3	7	6
12	4	2	1	4	2	1	2.141	0.6230	41	3	3	6	7
0	5	1	1	5	1	1	2.280	0.6683	42	4	4	8	7

Table 20. Modeling Results for an 8-3-8-1 Network.

Configuration Identification #	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3	λ_{PFE} value	Average p(BLUE Win) over all RED configurations	Ordinal Rank by p(BLUE Win)	Disparity of Sensors	Disparity of Influencers	Total Disparity	Robustness
1	6	1	1	1	6	1	1.899	0.1872	1	5	5	10	3
3	6	1	1	1	5	2	1.899	0.1996	2	5	4	9	3
14	5	2	1	1	1	6	1.899	0.2051	3	4	5	9	3
6	6	1	1	1	4	3	1.899	0.2085	4	5	3	8	3
35	4	3	1	1	1	6	1.899	0.2137	5	3	5	8	3
20	5	2	1	1	2	5	1.934	0.2637	6	4	4	8	4
26	5	2	1	1	3	4	1.968	0.2684	7	4	3	7	4
41	4	3	1	1	2	5	1.968	0.3056	8	3	4	7	4
25	5	2	1	1	4	3	2.000	0.3145	9	4	3	7	4
55	4	2	2	1	6	1	2.060	0.3184	10	2	5	7	4
4	6	1	1	2	1	5	2.060	0.3188	11	5	4	9	4
18	5	2	1	2	1	5	2.031	0.3269	12	4	4	8	4
13	5	2	1	1	6	1	2.060	0.3274	13	4	5	9	4
39	4	3	1	2	1	5	2.000	0.3325	14	3	4	7	4
19	5	2	1	1	5	2	2.031	0.3355	15	4	4	8	4
9	6	1	1	2	4	2	2.060	0.3457	16	5	2	7	4
11	6	1	1	2	3	3	2.060	0.3526	17	5	1	6	4
47	4	3	1	1	3	4	2.031	0.3829	18	3	3	6	5
67	3	2	3	1	6	1	2.060	0.3880	19	1	5	6	4
60	4	2	2	1	4	3	2.060	0.3932	20	2	3	5	5
45	4	3	1	3	1	4	2.088	0.3957	21	3	3	6	5
29	5	2	1	2	2	4	2.060	0.4047	22	4	2	6	5
50	4	3	1	2	2	4	2.060	0.4077	23	3	2	5	5
46	4	3	1	1	4	3	2.088	0.4094	24	3	3	6	5
57	4	2	2	1	5	2	2.060	0.4128	25	2	4	6	5
24	5	2	1	3	1	4	2.141	0.4274	26	4	3	7	5
58	4	2	2	2	1	5	2.115	0.4308	27	2	4	6	5
32	5	2	1	2	3	3	2.088	0.4402	28	4	1	5	5
69	3	2	3	1	5	2	2.088	0.4470	29	1	4	5	5
40	4	3	1	1	5	2	2.141	0.4487	30	3	4	7	5
7	6	1	1	3	1	4	2.190	0.4500	31	5	3	8	5
34	4	3	1	1	6	1	2.190	0.4585	32	3	5	8	5
10	6	1	1	3	3	2	2.190	0.4684	33	5	1	6	5
66	3	2	3	6	1	1	2.190	0.4692	34	1	5	6	5
28	5	2	1	2	4	2	2.115	0.4692	35	4	2	6	5
17	5	2	1	2	5	1	2.141	0.4705	36	4	4	8	5
43	4	3	1	4	1	3	2.166	0.4799	37	3	3	6	6
53	4	3	1	2	3	3	2.115	0.4825	38	3	1	4	6
65	4	2	2	2	3	3	2.115	0.4829	39	2	1	3	6

Table 20 (continued).

Configuration Identification #	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3	λ_{PFE} Value	Average p(BLUE Win) over all RED configurations	Ordinal Rank by p(BLUE Win)	Disparity of Sensors	Disparity of Influencers	Total Disparity	Robustness
72	3	2	3	1	4	3	2.115	0.4944	40	1	3	4	6
75	3	2	3	2	4	2	2.115	0.4979	41	1	2	3	6
52	4	3	1	3	2	3	2.141	0.5047	42	3	1	4	6
63	4	2	2	2	4	2	2.115	0.5081	43	2	2	4	6
23	5	2	1	3	4	1	2.213	0.5081	44	4	3	7	6
70	3	2	3	2	1	5	2.190	0.5098	45	1	4	5	6
30	5	2	1	3	3	2	2.190	0.5103	46	4	1	5	6
76	3	2	3	3	3	2	2.141	0.5115	47	1	1	2	7
31	5	2	1	3	2	3	2.166	0.5115	48	4	1	5	6
68	3	2	3	5	2	1	2.166	0.5265	49	1	4	5	6
61	4	2	2	3	1	4	2.166	0.5286	50	2	3	5	6
64	4	2	2	3	3	2	2.166	0.5346	51	2	1	3	7
71	3	2	3	4	3	1	2.141	0.5402	52	1	3	4	6
38	4	3	1	2	5	1	2.213	0.5427	53	3	4	7	6
33	4	3	1	6	1	1	2.300	0.5479	54	3	5	8	6
73	3	2	3	3	1	4	2.190	0.5496	55	1	3	4	7
49	4	3	1	2	4	2	2.166	0.5521	56	3	2	5	6
37	4	3	1	5	1	2	2.236	0.5526	57	3	4	7	6
51	4	3	1	3	3	2	2.190	0.5560	58	3	1	4	7
22	5	2	1	4	1	3	2.236	0.5594	59	4	3	7	6
77	3	2	3	3	2	3	2.166	0.5650	60	1	1	2	8
44	4	3	1	3	4	1	2.236	0.5739	61	3	3	6	7
5	6	1	1	4	3	1	2.300	0.5752	62	5	3	8	6
59	4	2	2	4	3	1	2.213	0.5765	63	2	3	5	7
54	4	2	2	6	1	1	2.300	0.5791	64	2	5	7	6
74	3	2	3	4	2	2	2.166	0.5795	65	1	2	3	7
62	4	2	2	4	2	2	2.213	0.5863	66	2	2	4	8
48	4	3	1	4	2	2	2.213	0.5923	67	3	2	5	7
16	5	2	1	5	1	2	2.231	0.6013	68	4	4	8	7
8	6	1	1	4	2	2	2.300	0.6047	69	5	2	7	6
15	5	2	1	5	2	1	2.340	0.6081	70	4	4	8	8
42	4	3	1	4	3	1	2.258	0.6162	71	3	3	6	8
56	4	2	2	5	2	1	2.258	0.6162	72	2	4	6	7
21	5	2	1	4	3	1	2.280	0.6201	73	4	3	7	7
27	5	2	1	4	2	2	2.258	0.6222	74	4	2	6	7
2	6	1	1	5	2	1	2.397	0.6312	75	5	4	9	7
36	4	3	1	5	2	1	2.280	0.6342	76	3	4	7	7
12	5	2	1	6	1	1	2.397	0.6402	77	4	5	9	7
0	6	1	1	6	1	1	2.483	0.6962	78	5	5	10	8

Table 21. Modeling Results for a 9-5-9-1 Network.

Configuration Identification #	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Sensors linked to Decider 4	Number of Sensors linked to Decider 5	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3	Number of Influencers linked to Decider 4	Number of Influencers linked to Decider 5	A_{PFE} value	Average p(BLUE Win) over all RED configurations	Ordinal Rank by p(BLUE Win)	Disparity of Sensors	Disparity of Influencers	Total Disparity	Robustness
1	5	1	1	1	1	1	5	1	1	1	1.899	0.2070	1	4	4	8	5
3	5	1	1	1	1	1	4	2	1	1	1.899	0.2530	2	4	3	7	5
14	4	2	1	1	1	1	1	5	1	1	1.899	0.2611	3	3	4	7	5
11	5	1	1	1	1	1	2	2	2	2	1.899	0.2758	4	4	1	5	5
21	4	2	1	1	1	1	1	4	2	1	1.899	0.2761	5	3	3	6	5
25	4	2	1	1	1	1	1	3	3	1	1.899	0.2779	6	3	2	5	5
6	5	1	1	1	1	1	3	3	1	1	1.899	0.2821	7	4	2	6	5
33	4	2	1	1	1	1	1	3	2	2	1.899	0.2993	8	3	2	5	5
55	3	2	2	1	1	1	1	1	5	1	1.899	0.3056	9	2	4	6	5
38	3	3	1	1	1	1	1	5	1	1	1.899	0.3088	10	2	4	6	5
8	5	1	1	1	1	1	3	2	2	1	1.899	0.3095	11	4	2	6	5
84	2	2	2	2	1	1	1	1	1	5	1.899	0.3291	12	1	4	5	5
4	5	1	1	1	1	2	1	4	1	1	2.031	0.3312	13	4	3	7	6
45	3	3	1	1	1	1	1	3	3	1	1.899	0.3333	14	2	2	4	5
63	3	2	2	1	1	1	1	1	4	2	1.899	0.3354	15	2	3	5	5
36	4	2	1	1	1	1	2	2	2	2	1.934	0.3365	16	3	1	4	6
20	4	2	1	1	1	1	2	4	1	1	1.934	0.3379	17	3	3	6	6
42	3	3	1	1	1	1	1	4	2	1	1.899	0.3446	18	2	3	5	5
9	5	1	1	1	1	2	3	2	1	1	2.031	0.3495	19	4	2	6	6
68	3	2	2	1	1	1	1	1	3	3	1.899	0.3575	20	2	2	4	5
19	4	2	1	1	1	1	4	2	1	1	2.000	0.3586	21	3	3	6	6
18	4	2	1	1	1	2	1	4	1	1	2.000	0.3593	22	3	3	6	6
32	4	2	1	1	1	1	2	3	2	1	1.934	0.3625	23	3	2	5	6
41	3	3	1	1	1	2	1	4	1	1	1.968	0.3663	24	2	3	5	6
59	3	2	2	1	1	2	1	1	4	1	1.968	0.3677	25	2	3	5	6
50	3	3	1	1	1	1	1	3	2	2	1.899	0.3688	26	2	2	4	5
35	4	2	1	1	1	2	1	2	2	2	2.000	0.3712	27	3	1	4	6
24	4	2	1	1	1	1	3	3	1	1	1.968	0.3716	28	3	2	5	6
86	2	2	2	2	1	4	1	1	1	2	2.088	0.3754	29	1	3	4	6
62	3	2	2	1	1	1	2	1	4	1	1.934	0.3765	30	2	3	5	6
52	3	3	1	1	1	2	1	2	2	2	1.968	0.3772	31	2	1	3	6
5	5	1	1	1	1	3	3	1	1	1	2.141	0.3782	32	4	2	6	7

Table 21 (continued).

Configuration Identification #	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Sensors linked to Decider 4	Number of Sensors linked to Decider 5	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3	Number of Influencers linked to Decider 4	Number of Influencers linked to Decider 5	A_{PFE} value	Average p(BLUE Win) over all RED configurations	Ordinal Rank by p(BLUE Win)	Disparity of Sensors	Disparity of Influencers	Total Disparity	Robustness
75	3	2	2	1	1	2	1	1	3	2	**1.968**	0.3811	**33**	2	2	**4**	**6**
10	5	1	1	1	1	2	2	2	2	1	**2.031**	0.3828	**34**	4	1	**5**	**6**
13	4	2	1	1	1	1	5	1	1	1	**2.031**	0.3842	**35**	3	4	**7**	**6**
49	3	3	1	1	1	2	1	3	2	1	**1.968**	0.3846	**36**	2	2	**4**	**6**
31	4	2	1	1	1	2	1	3	2	1	**2.000**	0.3954	**37**	3	2	**5**	**6**
71	3	2	2	1	1	3	1	1	2	2	**2.031**	0.3965	**38**	2	2	**4**	**7**
87	2	2	2	2	1	2	1	1	1	4	**1.968**	0.4018	**39**	1	3	**4**	**6**
67	3	2	2	1	1	1	3	1	3	1	**1.968**	0.4021	**40**	2	2	**4**	**6**
29	4	2	1	1	1	1	3	2	2	1	**1.968**	0.4021	**41**	3	2	**5**	**6**
54	3	2	2	1	1	1	5	1	1	1	**2.031**	0.4025	**42**	2	4	**6**	**6**
65	3	2	2	1	1	3	1	1	3	1	**2.031**	0.4025	**43**	2	2	**4**	**7**
83	2	2	2	2	1	5	1	1	1	1	**2.141**	0.4042	**44**	1	4	**5**	**6**
7	5	1	1	1	1	3	2	2	1	1	**2.141**	0.4105	**45**	4	2	**6**	**7**
44	3	3	1	1	1	3	1	3	1	1	**2.031**	0.4119	**46**	2	2	**4**	**7**
23	4	2	1	1	1	3	1	3	1	1	**2.088**	0.4119	**47**	3	2	**5**	**7**
79	3	2	2	1	1	1	2	1	3	2	**1.934**	0.4123	**48**	2	2	**4**	**6**
74	3	2	2	1	1	2	2	1	3	1	**2.000**	0.4175	**49**	2	2	**4**	**7**
77	3	2	2	1	1	1	3	1	2	2	**1.968**	0.4232	**50**	2	2	**4**	**6**
89	2	2	2	2	1	3	1	1	1	3	**2.031**	0.4235	**51**	1	2	**3**	**6**
47	3	3	1	1	1	3	1	2	2	1	**2.031**	0.4260	**52**	2	4	**6**	**7**
37	3	3	1	1	1	5	1	1	1	1	**2.141**	0.4260	**53**	2	2	**4**	**7**
40	3	3	1	1	1	4	1	2	1	1	**2.088**	0.4298	**54**	2	3	**5**	**7**
27	4	2	1	1	1	3	1	2	2	1	**2.088**	0.4298	**55**	3	2	**5**	**7**
81	3	2	2	1	1	2	2	1	2	2	**2.000**	0.4323	**56**	2	1	**3**	**7**
85	2	2	2	2	1	4	2	1	1	1	**2.115**	0.4333	**57**	1	3	**4**	**7**
30	4	2	1	1	1	2	2	3	1	1	**2.031**	0.4340	**58**	3	2	**5**	**7**
16	4	2	1	1	1	4	1	2	1	1	**2.166**	0.4375	**59**	3	3	**6**	**8**
78	3	2	2	1	1	1	2	2	3	1	**1.968**	0.4407	**60**	2	2	**4**	**7**
34	4	2	1	1	1	2	2	2	2	1	**2.031**	0.4432	**61**	3	1	**4**	**7**
82	3	2	2	1	1	1	2	2	2	2	**1.968**	0.4470	**62**	2	1	**3**	**7**
61	3	2	2	1	1	1	4	1	2	1	**2.000**	0.4474	**63**	2	3	**5**	**6**
53	3	2	2	1	1	5	1	1	1	1	**2.141**	0.4495	**64**	2	4	**6**	**7**

Table 21 (continued).

Configuration Identification #	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Sensors linked to Decider 4	Number of Sensors linked to Decider 5	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3	Number of Influencers linked to Decider 4	Number of Influencers linked to Decider 5	A_{PFE} value	Average p(BLUE Win) over all RED configurations	Ordinal Rank by p(BLUE Win)	Disparity of Sensors	Disparity of Influencers	Total Disparity	Robustness
91	2	2	2	2	1	3	2	1	1	2	**2.060**	0.4519	**65**	1	2	**3**	**7**
28	4	2	1	1	1	2	3	2	1	1	**2.060**	0.4558	**66**	3	2	**5**	**7**
17	4	2	1	1	1	2	4	1	1	1	**2.088**	0.4586	**67**	3	3	**6**	**7**
48	3	3	1	1	1	2	2	3	1	1	**2.031**	0.4621	**68**	2	2	**4**	**7**
73	3	2	2	1	1	2	3	1	2	1	**2.031**	0.4642	**69**	2	2	**4**	**7**
57	3	2	2	1	1	4	1	1	2	1	**2.088**	0.4681	**70**	2	3	**5**	**7**
58	3	2	2	1	1	2	4	1	1	1	**2.060**	0.4709	**71**	2	3	**5**	**7**
92	2	2	2	2	1	2	2	1	1	3	**2.000**	0.4737	**72**	4	3	**7**	**8**
2	5	1	1	1	1	4	2	1	1	1	**2.236**	0.4737	**73**	1	2	**3**	**7**
51	3	3	1	1	1	2	2	2	2	1	**2.031**	0.4747	**74**	2	1	**3**	**7**
70	3	2	2	1	1	3	2	1	2	1	**2.060**	0.4775	**75**	2	2	**4**	**8**
26	4	2	1	1	1	3	2	2	1	1	**2.115**	0.4793	**76**	3	2	**5**	**8**
22	4	2	1	1	1	3	3	1	1	1	**2.141**	0.4796	**77**	3	2	**5**	**8**
60	3	2	2	1	1	1	4	2	1	1	**2.031**	0.4800	**78**	2	3	**5**	**7**
76	3	2	2	1	1	1	3	2	2	1	**2.000**	0.4825	**79**	2	2	**4**	**7**
94	2	2	2	2	1	2	2	2	1	2	**2.031**	0.4860	**80**	1	1	**2**	**8**
15	4	2	1	1	1	4	2	1	1	1	**2.190**	0.4888	**81**	3	3	**6**	**9**
12	4	2	1	1	1	5	1	1	1	1	**2.236**	0.4891	**82**	3	4	**7**	**8**
56	3	2	2	1	1	4	2	1	1	1	**2.115**	0.4902	**83**	2	3	**5**	**8**
64	3	2	2	1	1	3	3	1	1	1	**2.088**	0.4916	**84**	2	2	**4**	**8**
88	2	2	2	2	1	3	3	1	1	1	**2.088**	0.4989	**85**	1	2	**3**	**7**
46	3	3	1	1	1	3	2	2	1	1	**2.088**	0.5042	**86**	2	2	**4**	**8**
66	3	2	2	1	1	1	3	3	1	1	**2.031**	0.5077	**87**	2	2	**4**	**7**
0	5	1	1	1	1	5	1	1	1	1	**2.321**	0.5081	**88**	4	4	**8**	**9**
80	3	2	2	1	1	2	2	2	2	1	**2.031**	0.5095	**89**	2	1	**3**	**8**
90	2	2	2	2	1	3	2	2	1	1	**2.088**	0.5161	**90**	1	2	**3**	**8**
72	3	2	2	1	1	2	3	2	1	1	**2.060**	0.5168	**91**	2	2	**4**	**8**
69	3	2	2	1	1	3	2	2	1	1	**2.088**	0.5172	**92**	2	2	**4**	**9**
39	3	3	1	1	1	4	2	1	1	1	**2.141**	0.5249	**93**	2	3	**5**	**8**
93	2	2	2	2	1	2	2	2	2	1	**2.031**	0.5425	**94**	1	1	**2**	**9**
43	3	3	1	1	1	3	3	1	1	1	**2.141**	0.5684	**95**	2	2	**4**	**9**

Table 22. Modeling Results for a 9-5-9-1 Network (not including ties).

Configuration Identification #	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Sensors linked to Decider 4	Number of Sensors linked to Decider 5	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3	Number of Influencers linked to Decider 4	Number of Influencers linked to Decider 5	Λ_{PFE} value	Average p(BLUE Win) over all RED configurations	Ordinal Rank by p(BLUE Win)	Disparity of Sensors	Disparity of Influencers	Total Disparity	Robustness
1	5	1	1	1	1	1	5	1	1	1	1.899	0.2215	1	4	4	8	5
3	5	1	1	1	1	1	4	2	1	1	1.899	0.2713	2	4	3	7	5
14	4	2	1	1	1	1	1	5	1	1	1.899	0.2802	3	3	4	7	5
11	5	1	1	1	1	1	2	2	2	2	1.899	0.2944	4	4	1	5	5
21	4	2	1	1	1	1	1	4	2	1	1.899	0.2947	5	3	3	6	5
25	4	2	1	1	1	1	1	3	3	1	1.899	0.2961	6	3	2	5	5
6	5	1	1	1	1	1	3	3	1	1	1.899	0.3019	7	4	2	6	5
33	4	2	1	1	1	1	1	3	2	2	1.899	0.3179	8	3	2	5	5
55	3	2	2	1	1	1	1	1	5	1	1.899	0.3258	9	2	4	6	5
38	3	3	1	1	1	1	1	5	1	1	1.899	0.3308	10	2	4	6	5
8	5	1	1	1	1	1	3	2	2	1	1.899	0.3332	11	4	2	6	5
84	2	2	2	2	1	1	1	1	1	5	1.899	0.3492	12	1	4	5	5
45	3	3	1	1	1	1	1	3	3	1	1.899	0.3554	13	2	2	4	5
36	4	2	1	1	1	1	2	2	2	2	1.934	0.3586	14	3	1	4	6
63	3	2	2	1	1	1	1	1	4	2	1.899	0.3592	15	2	3	5	5
20	4	2	1	1	1	1	2	4	1	1	1.934	0.3594	16	3	3	6	6
4	5	1	1	1	1	2	1	4	1	1	2.031	0.3620	17	4	3	7	6
42	3	3	1	1	1	1	1	4	2	1	1.899	0.3644	18	2	3	5	5
68	3	2	2	1	1	1	1	1	3	3	1.899	0.3774	19	2	2	4	5
9	5	1	1	1	1	2	3	2	1	1	2.031	0.3794	20	4	2	6	6
19	4	2	1	1	1	1	4	2	1	1	2.000	0.3836	21	3	3	6	6
32	4	2	1	1	1	1	2	3	2	1	1.934	0.3885	22	3	2	5	6
50	3	3	1	1	1	1	1	3	2	2	1.899	0.3888	23	2	2	4	5
59	3	2	2	1	1	2	1	1	4	1	1.968	0.3917	24	2	3	5	6
18	4	2	1	1	1	2	1	4	1	1	2.000	0.3945	25	3	3	6	6
41	3	3	1	1	1	2	1	4	1	1	1.968	0.3958	26	2	3	5	6
24	4	2	1	1	1	1	3	3	1	1	1.968	0.3968	27	3	2	5	6
35	4	2	1	1	1	2	1	2	2	2	2.000	0.3969	28	3	1	4	6
62	3	2	2	1	1	1	2	1	4	1	1.934	0.3996	29	2	3	5	6
52	3	3	1	1	1	2	1	2	2	2	1.968	0.4005	30	2	1	3	6
75	3	2	2	1	1	2	1	1	3	2	1.968	0.4066	31	2	2	4	6
86	2	2	2	2	1	4	1	1	1	2	2.088	0.4068	32	1	3	4	6

Table 22 (continued).

Configuration Identification #	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Sensors linked to Decider 4	Number of Sensors linked to Decider 5	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3	Number of Influencers linked to Decider 4	Number of Influencers linked to Decider 5	Λ_{PFE} Value	Average p(BLUE Win) over all RED configurations	Ordinal Rank by p(BLUE Win)	Disparity of Sensors	Disparity of Influencers	Total Disparity	Robustness
13	4	2	1	1	1	1	5	1	1	1	**2.031**	0.4082	**33**	3	4	**7**	**6**
10	5	1	1	1	1	2	2	2	2	1	**2.031**	0.4133	**34**	4	1	**5**	**6**
49	3	3	1	1	1	2	1	3	2	1	**1.968**	0.4136	**35**	2	2	**4**	**6**
87	2	2	2	2	1	2	1	1	1	4	**1.968**	0.4278	**36**	1	3	**4**	**6**
5	5	1	1	1	1	3	3	1	1	1	**2.141**	0.4291	**37**	4	2	**6**	**7**
67	3	2	2	1	1	1	3	1	3	1	**1.968**	0.4299	**38**	2	2	**4**	**6**
71	3	2	2	1	1	3	1	1	2	2	**2.031**	0.4303	**39**	2	2	**4**	**7**
31	4	2	1	1	1	2	1	3	2	1	**2.000**	0.4306	**40**	3	2	**5**	**6**
29	4	2	1	1	1	1	3	2	2	1	**1.968**	0.4318	**41**	3	2	**5**	**6**
54	3	2	2	1	1	1	5	1	1	1	**2.031**	0.4319	**42**	2	4	**6**	**6**
65	3	2	2	1	1	3	1	1	3	1	**2.031**	0.4348	**43**	2	2	**4**	**7**
79	3	2	2	1	1	1	2	1	3	2	**1.934**	0.4369	**44**	2	2	**4**	**6**
83	2	2	2	2	1	5	1	1	1	1	**2.141**	0.4416	**45**	1	4	**5**	**6**
44	3	3	1	1	1	3	1	3	1	1	**2.031**	0.4463	**46**	2	2	**4**	**7**
74	3	2	2	1	1	2	2	1	3	1	**2.000**	0.4464	**47**	2	2	**4**	**7**
23	4	2	1	1	1	3	1	3	1	1	**2.088**	0.4508	**48**	3	2	**5**	**7**
77	3	2	2	1	1	1	3	1	2	2	**1.968**	0.4522	**49**	2	2	**4**	**6**
89	2	2	2	2	1	3	1	1	1	3	**2.031**	0.4553	**50**	1	2	**3**	**6**
47	3	3	1	1	1	3	1	2	2	1	**2.031**	0.4634	**51**	2	2	**4**	**7**
81	3	2	2	1	1	2	2	1	2	2	**2.000**	0.4638	**52**	4	2	**6**	**7**
7	5	1	1	1	1	3	2	2	1	1	**2.141**	0.4660	**53**	2	1	**3**	**7**
78	3	2	2	1	1	1	2	2	3	1	**1.968**	0.4663	**54**	2	2	**4**	**7**
85	2	2	2	2	1	4	2	1	1	1	**2.115**	0.4680	**55**	1	3	**4**	**7**
27	4	2	1	1	1	3	1	2	2	1	**2.088**	0.4705	**56**	3	2	**5**	**7**
30	4	2	1	1	1	2	2	3	1	1	**2.031**	0.4717	**57**	3	2	**5**	**7**
40	3	3	1	1	1	4	1	2	1	1	**2.088**	0.4746	**58**	2	3	**5**	**7**
82	3	2	2	1	1	1	2	2	2	2	**1.968**	0.4748	**59**	2	1	**3**	**7**
61	3	2	2	1	1	1	4	1	2	1	**2.000**	0.4757	**60**	2	3	**5**	**6**
37	3	3	1	1	1	5	1	1	1	1	**2.141**	0.4824	**61**	2	4	**6**	**7**
34	4	2	1	1	1	2	2	2	2	1	**2.031**	0.4833	**62**	3	1	**4**	**7**
91	2	2	2	2	1	3	2	1	1	2	**2.060**	0.4872	**63**	1	2	**3**	**7**
16	4	2	1	1	1	4	1	2	1	1	**2.166**	0.4932	**64**	3	3	**6**	**8**

Table 22 (continued).

Configuration identification #	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Sensors linked to Decider 4	Number of Sensors linked to Decider 5	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3	Number of Influencers linked to Decider 4	Number of Influencers linked to Decider 5	λ_{PFE} value	Average p(BLUE Win) over all RED configurations	Ordinal Rank by p(BLUE Win)	Disparity of Sensors	Disparity of Influencers	Total Disparity	Robustness
73	3	2	2	1	1	2	3	1	2	1	2.031	0.4954	65	2	2	4	7
48	3	3	1	1	1	2	2	3	1	1	2.031	0.4956	66	2	2	4	7
28	4	2	1	1	1	2	3	2	1	1	2.060	0.4956	67	3	2	5	7
53	3	2	2	1	1	5	1	1	1	1	2.141	0.4980	68	2	4	6	7
17	4	2	1	1	1	2	4	1	1	1	2.088	0.4989	69	3	3	6	7
92	2	2	2	2	1	2	2	1	1	3	2.000	0.5032	70	1	2	3	7
58	3	2	2	1	1	2	4	1	1	1	2.060	0.5047	71	2	3	5	7
51	3	3	1	1	1	2	2	2	2	1	2.031	0.5077	72	2	3	5	7
60	3	2	2	1	1	1	4	2	1	1	2.031	0.5108	73	2	1	3	7
76	3	2	2	1	1	1	3	2	2	1	2.000	0.5110	74	2	2	4	7
57	3	2	2	1	1	4	1	1	2	1	2.088	0.5159	75	2	3	5	7
70	3	2	2	1	1	3	2	1	2	1	2.060	0.5187	76	2	2	4	8
94	2	2	2	2	1	2	2	2	1	2	2.031	0.5189	77	1	1	2	8
22	4	2	1	1	1	3	3	1	1	1	2.141	0.5243	78	3	2	5	8
26	4	2	1	1	1	3	2	2	1	1	2.115	0.5276	79	3	2	5	8
64	3	2	2	1	1	3	3	1	1	1	2.088	0.5298	80	2	2	4	8
88	2	2	2	2	1	3	3	1	1	1	2.088	0.5331	81	1	2	3	7
56	3	2	2	1	1	4	2	1	1	1	2.115	0.5356	82	2	3	5	8
66	3	2	2	1	1	1	3	3	1	1	2.031	0.5382	83	2	2	4	7
2	5	1	1	1	1	4	2	1	1	1	2.236	0.5415	84	4	3	7	8
80	3	2	2	1	1	2	2	2	2	1	2.031	0.5452	85	2	1	3	8
46	3	3	1	1	1	3	2	2	1	1	2.088	0.5456	86	2	2	4	8
15	4	2	1	1	1	4	2	1	1	1	2.190	0.5504	87	3	3	6	9
72	3	2	2	1	1	2	3	2	1	1	2.060	0.5515	88	2	2	4	8
90	2	2	2	2	1	3	2	2	1	1	2.088	0.5515	89	1	2	3	8
12	4	2	1	1	1	5	1	1	1	1	2.236	0.5587	90	3	4	7	8
69	3	2	2	1	1	3	2	2	1	1	2.088	0.5625	91	2	2	4	9
39	3	3	1	1	1	4	2	1	1	1	2.141	0.5705	92	2	3	5	8
93	2	2	2	2	1	2	2	2	2	1	2.031	0.5710	93	1	1	2	9
0	5	1	1	1	1	5	1	1	1	1	2.321	0.5944	94	4	4	8	9
43	3	3	1	1	1	3	3	1	1	1	2.141	0.6109	95	2	2	4	9

Table 23. Modeling Results for a 9-5-9-1 Network (Ranked by λ_{PFE} value).

Configuration identification #	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Sensors linked to Decider 4	Number of Sensors linked to Decider 5	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3	Number of Influencers linked to Decider 4	Number of Influencers linked to Decider 5	λ_{PFE} value	Average p(BLUE Win) over all RED configurations	Ordinal Rank by λ_{PFE} value	Disparity of Sensors	Disparity of Influencers	Total Disparity	Robustness
1	5	1	1	1	1	1	5	1	1	1	1.899	0.2070	1	4	4	8	5
3	5	1	1	1	1	1	4	2	1	1	1.899	0.2530	2	4	3	7	5
14	4	2	1	1	1	1	1	5	1	1	1.899	0.2611	3	3	4	7	5
11	5	1	1	1	1	1	2	2	2	2	1.899	0.2758	4	4	1	5	5
21	4	2	1	1	1	1	1	4	2	1	1.899	0.2761	5	3	3	6	5
25	4	2	1	1	1	1	1	3	3	1	1.899	0.2779	6	3	2	5	5
6	5	1	1	1	1	1	3	3	1	1	1.899	0.2821	7	4	2	6	5
33	4	2	1	1	1	1	1	3	2	2	1.899	0.2993	8	3	2	5	5
55	3	2	2	1	1	1	1	1	5	1	1.899	0.3056	9	2	4	6	5
38	3	3	1	1	1	1	1	5	1	1	1.899	0.3088	10	2	4	6	5
8	5	1	1	1	1	1	3	2	2	1	1.899	0.3095	11	4	2	6	5
84	2	2	2	2	1	1	1	1	1	5	1.899	0.3291	12	1	4	5	5
45	3	3	1	1	1	1	1	3	3	1	1.899	0.3333	13	2	2	4	5
63	3	2	2	1	1	1	1	1	4	2	1.899	0.3354	14	2	3	5	5
42	3	3	1	1	1	1	1	4	2	1	1.899	0.3446	15	2	3	5	5
68	3	2	2	1	1	1	1	1	3	3	1.899	0.3575	16	2	2	4	5
50	3	3	1	1	1	1	1	3	2	2	1.899	0.3688	17	2	2	4	5
36	4	2	1	1	1	1	2	2	2	2	1.934	0.3365	18	3	1	4	6
20	4	2	1	1	1	1	2	4	1	1	1.934	0.3379	19	3	3	6	6
32	4	2	1	1	1	1	2	3	2	1	1.934	0.3625	20	3	2	5	6
62	3	2	2	1	1	1	2	1	4	1	1.934	0.3765	21	2	3	5	6
79	3	2	2	1	1	1	2	1	3	2	1.934	0.4123	22	2	2	4	6
41	3	3	1	1	1	2	1	4	1	1	1.968	0.3663	23	2	3	5	6
59	3	2	2	1	1	2	1	1	4	1	1.968	0.3677	24	2	3	5	6
24	4	2	1	1	1	1	3	3	1	1	1.968	0.3716	25	3	2	5	6
52	3	3	1	1	1	2	1	2	2	2	1.968	0.3772	26	2	1	3	6
75	3	2	2	1	1	2	1	1	3	2	1.968	0.3811	27	2	2	4	6
49	3	3	1	1	1	2	1	3	2	1	1.968	0.3846	28	2	2	4	6
87	2	2	2	2	1	2	1	1	1	4	1.968	0.4018	29	1	3	4	6
67	3	2	2	1	1	1	3	1	3	1	1.968	0.4021	30	2	2	4	6
29	4	2	1	1	1	1	3	2	2	1	1.968	0.4021	31	3	2	5	6
77	3	2	2	1	1	1	3	1	2	2	1.968	0.4232	32	2	2	4	6

Table 23 (continued).

Configuration Identification #	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Sensors linked to Decider 4	Number of Sensors linked to Decider 5	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3	Number of Influencers linked to Decider 4	Number of Influencers linked to Decider 5	A_{PFE} value	Average p(BLUE Win) over all RED configurations	Ordinal Rank by A_{PFE} value	Disparity of Sensors	Disparity of Influencers	Total Disparity	Robustness
78	3	2	2	1	1	1	2	2	3	1	**1.968**	0.4407	**33**	2	2	**4**	**7**
82	3	2	2	1	1	1	2	2	2	2	**1.968**	0.4470	**34**	2	1	**3**	**7**
19	4	2	1	1	1	1	4	2	1	1	**2.000**	0.3586	**35**	3	3	**6**	**6**
18	4	2	1	1	1	2	1	4	1	1	**2.000**	0.3593	**36**	3	3	**6**	**6**
35	4	2	1	1	1	2	1	2	2	2	**2.000**	0.3712	**37**	3	1	**4**	**6**
31	4	2	1	1	1	2	1	3	2	1	**2.000**	0.3954	**38**	3	2	**5**	**6**
74	3	2	2	1	1	2	2	1	3	1	**2.000**	0.4175	**39**	2	2	**4**	**7**
81	3	2	2	1	1	2	2	1	2	2	**2.000**	0.4323	**40**	2	1	**3**	**7**
61	3	2	2	1	1	1	4	1	2	1	**2.000**	0.4474	**41**	2	3	**5**	**6**
92	2	2	2	2	1	2	2	1	1	3	**2.000**	0.4737	**42**	2	2	**4**	**7**
76	3	2	2	1	1	1	3	2	2	1	**2.000**	0.4825	**43**	2	2	**4**	**7**
4	5	1	1	1	1	2	1	4	1	1	**2.031**	0.3312	**44**	4	3	**7**	**6**
9	5	1	1	1	1	2	3	2	1	1	**2.031**	0.3495	**45**	4	2	**6**	**6**
10	5	1	1	1	1	2	2	2	2	1	**2.031**	0.3828	**46**	4	1	**5**	**6**
13	4	2	1	1	1	1	5	1	1	1	**2.031**	0.3842	**47**	3	4	**7**	**6**
71	3	2	2	1	1	3	1	1	2	2	**2.031**	0.3965	**48**	2	2	**4**	**7**
54	3	2	2	1	1	1	5	1	1	1	**2.031**	0.4025	**49**	2	4	**6**	**6**
65	3	2	2	1	1	3	1	1	3	1	**2.031**	0.4025	**50**	2	2	**4**	**7**
44	3	3	1	1	1	3	1	3	1	1	**2.031**	0.4119	**51**	2	2	**4**	**7**
89	2	2	2	2	1	3	1	1	1	3	**2.031**	0.4235	**52**	1	2	**3**	**6**
47	3	3	1	1	1	3	1	2	2	1	**2.031**	0.4260	**53**	3	2	**5**	**7**
30	4	2	1	1	1	2	2	3	1	1	**2.031**	0.4340	**54**	3	2	**5**	**7**
34	4	2	1	1	1	2	2	2	2	1	**2.031**	0.4432	**55**	3	1	**4**	**7**
48	3	3	1	1	1	2	2	3	1	1	**2.031**	0.4621	**56**	2	2	**4**	**7**
73	3	2	2	1	1	2	3	1	2	1	**2.031**	0.4642	**57**	2	2	**4**	**7**
51	3	3	1	1	1	2	2	2	2	1	**2.031**	0.4747	**58**	2	1	**3**	**7**
60	3	2	2	1	1	1	4	2	1	1	**2.031**	0.4800	**59**	2	3	**5**	**7**
94	2	2	2	2	1	2	2	2	1	2	**2.031**	0.4860	**60**	1	1	**2**	**8**
66	3	2	2	1	1	1	3	3	1	1	**2.031**	0.5077	**61**	2	2	**4**	**7**
80	3	2	2	1	1	2	2	2	2	1	**2.031**	0.5095	**62**	2	1	**3**	**8**
93	2	2	2	2	1	2	2	2	2	1	**2.031**	0.5425	**63**	1	1	**2**	**9**
91	2	2	2	2	1	3	2	1	1	2	**2.060**	0.4519	**64**	1	2	**3**	**7**

Table 23 (continued).

Configuration Identification #	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Sensors linked to Decider 4	Number of Sensors linked to Decider 5	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3	Number of Influencers linked to Decider 4	Number of Influencers linked to Decider 5	Λ_{PFE} Value	Average p(BLUE Win) over all RED configurations	Ordinal Rank by Λ_{PFE} value	Disparity of Sensors	Disparity of Influencers	Total Disparity	Robustness
28	4	2	1	1	1	2	3	2	1	1	**2.060**	0.4558	**65**	3	2	**5**	**7**
58	3	2	2	1	1	2	4	1	1	1	**2.060**	0.4709	**66**	2	3	**5**	**7**
70	3	2	2	1	1	3	2	1	2	1	**2.060**	0.4775	**67**	2	2	**4**	**8**
72	3	2	2	1	1	2	3	2	1	1	**2.060**	0.5168	**68**	2	2	**4**	**8**
86	2	2	2	2	1	4	1	1	1	2	**2.088**	0.3754	**69**	1	3	**4**	**6**
23	4	2	1	1	1	3	1	3	1	1	**2.088**	0.4119	**70**	3	2	**5**	**7**
40	3	3	1	1	1	4	1	2	1	1	**2.088**	0.4298	**71**	2	3	**5**	**7**
27	4	2	1	1	1	3	1	2	2	1	**2.088**	0.4298	**72**	3	2	**5**	**7**
17	4	2	1	1	1	2	4	1	1	1	**2.088**	0.4586	**73**	3	3	**6**	**7**
57	3	2	2	1	1	4	1	1	2	1	**2.088**	0.4681	**74**	2	3	**5**	**7**
64	3	2	2	1	1	3	3	1	1	1	**2.088**	0.4916	**75**	2	2	**4**	**8**
88	2	2	2	2	1	3	3	1	1	1	**2.088**	0.4989	**76**	1	2	**3**	**7**
46	3	3	1	1	1	3	2	2	1	1	**2.088**	0.5042	**77**	2	2	**4**	**8**
90	2	2	2	2	1	3	2	2	1	1	**2.088**	0.5161	**78**	1	2	**3**	**8**
69	3	2	2	1	1	3	2	2	1	1	**2.088**	0.5172	**79**	2	2	**4**	**9**
85	2	2	2	2	1	4	2	1	1	1	**2.115**	0.4333	**80**	1	3	**4**	**7**
26	4	2	1	1	1	3	2	2	1	1	**2.115**	0.4793	**81**	3	2	**5**	**8**
56	3	2	2	1	1	4	2	1	1	1	**2.115**	0.4902	**82**	2	3	**5**	**8**
5	5	1	1	1	1	3	3	1	1	1	**2.141**	0.3782	**83**	4	2	**6**	**7**
83	2	2	2	2	1	5	1	1	1	1	**2.141**	0.4042	**84**	1	4	**5**	**6**
7	5	1	1	1	1	3	2	2	1	1	**2.141**	0.4105	**85**	4	2	**6**	**7**
37	3	3	1	1	1	5	1	1	1	1	**2.141**	0.4260	**86**	4	2	**6**	**7**
53	3	2	2	1	1	5	1	1	1	1	**2.141**	0.4495	**87**	2	4	**6**	**7**
22	4	2	1	1	1	3	3	1	1	1	**2.141**	0.4796	**88**	3	2	**5**	**8**
39	3	3	1	1	1	4	2	1	1	1	**2.141**	0.5249	**89**	2	3	**5**	**8**
43	3	3	1	1	1	3	3	1	1	1	**2.141**	0.5684	**90**	2	2	**4**	**9**
16	4	2	1	1	1	4	1	2	1	1	**2.166**	0.4375	**91**	3	3	**6**	**8**
15	4	2	1	1	1	4	2	1	1	1	**2.190**	0.4888	**92**	3	3	**6**	**9**
2	5	1	1	1	1	4	2	1	1	1	**2.236**	0.4737	**93**	3	3	**6**	**9**
12	4	2	1	1	1	5	1	1	1	1	**2.236**	0.4891	**94**	3	4	**7**	**8**
0	5	1	1	1	1	5	1	1	1	1	**2.321**	0.5081	**95**	4	4	**8**	**9**

Table 24: Modeling Results for all 9-5-9-1 configurations with a λ_{PFE} value of 2.031.

Configuration Identification #	Number of Sensors linked to Decider 1	Number of Sensors linked to Decider 2	Number of Sensors linked to Decider 3	Number of Sensors linked to Decider 4	Number of Sensors linked to Decider 5	Number of Influencers linked to Decider 1	Number of Influencers linked to Decider 2	Number of Influencers linked to Decider 3	Number of Influencers linked to Decider 4	Number of Influencers linked to Decider 5	λ_{PFE} value	Average p(BLUE Win) over all RED configurations	Average p(BLUE Win) over all RED configurations (minus ties)	Disparity of Sensors	Disparity of Influencers	Total Disparity	Robustness
4	5	1	1	1	1	2	1	4	1	1	**2.031**	0.3312	0.3620	4	3	**7**	**6**
9	5	1	1	1	1	2	3	2	1	1	**2.031**	0.3495	0.3794	4	2	**6**	**6**
13	4	2	1	1	1	1	5	1	1	1	**2.031**	0.3842	0.4082	3	4	**7**	**6**
10	5	1	1	1	1	2	2	2	2	1	**2.031**	0.3828	0.4133	4	1	**5**	**6**
71	3	2	2	1	1	3	1	1	2	2	**2.031**	0.3965	0.4303	2	2	**4**	**7**
54	3	2	2	1	1	1	5	1	1	1	**2.031**	0.4025	0.4319	2	4	**6**	**6**
65	3	2	2	1	1	3	1	1	3	1	**2.031**	0.4025	0.4348	2	2	**4**	**7**
44	3	3	1	1	1	3	1	3	1	1	**2.031**	0.4119	0.4463	2	2	**4**	**7**
89	2	2	2	2	1	3	1	1	1	3	**2.031**	0.4235	0.4553	1	2	**3**	**6**
47	3	3	1	1	1	3	1	2	2	1	**2.031**	0.4260	0.4634	2	2	**4**	**7**
30	4	2	1	1	1	2	2	3	1	1	**2.031**	0.4340	0.4717	3	2	**5**	**7**
34	4	2	1	1	1	2	2	2	2	1	**2.031**	0.4432	0.4833	3	1	**4**	**7**
73	3	2	2	1	1	2	3	1	2	1	**2.031**	0.4642	0.4954	2	2	**4**	**7**
48	3	3	1	1	1	2	2	3	1	1	**2.031**	0.4621	0.4956	2	2	**4**	**7**
51	3	3	1	1	1	2	2	2	2	1	**2.031**	0.4747	0.5077	2	3	**5**	**7**
60	3	2	2	1	1	1	4	2	1	1	**2.031**	0.4800	0.5108	2	1	**3**	**7**
94	2	2	2	2	1	2	2	2	1	2	**2.031**	0.4860	0.5189	1	1	**2**	**8**
66	3	2	2	1	1	1	3	3	1	1	**2.031**	0.5077	0.5382	2	2	**4**	**7**
80	3	2	2	1	1	2	2	2	2	1	**2.031**	0.5095	0.5452	2	1	**3**	**8**
93	2	2	2	2	1	2	2	2	2	1	**2.031**	0.5425	0.5710	1	1	**2**	**9**

APPENDIX E

REGRESSION RESULTS

Table 25. Regression Results for a 7-3-7-1 Network.

Regression Statistics	
Multiple R	0.946487911
R Square	0.895839365
Adjusted R Square	0.893235349
Standard Error	0.037481712
Observations	42

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	0.483309524	0.483309524	344.0222363	2.9851E-21
Residual	40	0.056195149	0.001404879		
Total	41	0.539504673			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	-1.578048553	0.109974571	-14.34921309	2.29505E-17	-1.800315449	-1.355781656
X Variable 1	1.016281894	0.054792477	18.54783643	2.9851E-21	0.905542169	1.127021619

Table 26. Regression Results for an 8-3-8-1 Network.

Regression Statistics	
Multiple R	0.93597811
R Square	0.876055022
Adjusted R Square	0.874424167
Standard Error	0.043051289
Observations	78

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	0.995607969	0.995607969	537.175308	3.39433E-36
Residual	76	0.140859426	0.001853414		
Total	77	1.136467396			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	-1.563275855	0.087779687	-17.80908442	7.63064E-29	-1.738104251	-1.38844746
X Variable 1	0.948403389	0.040919948	23.17704269	3.39433E-36	0.86690425	1.029902528

Table 27. Regression Results for a 9-5-9-1 Network.

Regression Statistics	
Multiple R	0.720469333
R Square	0.519076059
Adjusted R Square	0.513904834
Standard Error	0.051037801
Observations	95

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	0.261469758	0.261469758	100.3777717	1.86779E-16
Residual	93	0.242251717	0.002604857		
Total	94	0.503721475			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	-0.773582375	0.118392171	-6.534066951	3.3772E-09	-1.008685758	-0.538478991
X Variable 1	0.58608456	0.058498066	10.01887078	1.86779E-16	0.469918998	0.702250122

Table 28. Regression Results for a 9-5-9-1 Network (minus ties).

Regression Statistics	
Multiple R	0.788156947
R Square	0.621191373
Adjusted R Square	0.617118162
Standard Error	0.050490392
Observations	95

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	0.388781853	0.388781853	152.5065526	2.59237E-21
Residual	93	0.237083009	0.00254928		
Total	94	0.625864861			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	-1.000220931	0.117122348	-8.539966551	2.49582E-13	-1.232802696	-0.767639166
X Variable 1	0.714665043	0.05787064	12.34935434	2.59237E-21	0.599745423	0.829584663

Table 29. Two-Variable Regression Results for a 9-5-9-1 Network (minus ties).

Regression Statistics	
Multiple R	0.922019216
R Square	0.850119434
Adjusted R Square	0.846861161
Standard Error	0.031931479
Observations	95

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	2	0.532059882	0.266029941	260.911037	1.21419E-38
Residual	92	0.09380498	0.001019619		
Total	94	0.625864861			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	-0.161260669	0.102447289	-1.574084292	0.11890076	-0.364729824	0.042208486
λ_{PFE} value	0.099672387	0.063490214	1.569885814	0.119874799	-0.026424656	0.22576943
Robustness value	0.061285406	0.00516995	11.85415947	3.1785E-20	0.05101744	0.071553372

Table 30. Two-Variable Regression Results for a 9-5-9-1 Network.

Regression Statistics	
Multiple R	0.897294287
R Square	0.805137037
Adjusted R Square	0.800900886
Standard Error	0.032663745
Observations	95

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	2	0.405564816	0.202782408	190.0633305	2.12592E-33
Residual	92	0.098156659	0.00106692		
Total	94	0.503721475			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	0.067766532	0.104796652	0.646647875	0.519469884	-0.14036866	0.275901724
λ_{PFE} value	-0.030659072	0.064946198	-0.472068781	0.637995484	-0.159647824	0.098329679
Robustness value	0.061459895	0.005288509	11.62140329	9.56346E-20	0.05095646	0.071963329

APPENDIX F

METRICS FOR A 7-3-7-1 NETWORK

The values of the network metrics depicted in Table 31 were calculated as follows:

- *Number of nodes* (N) = the total number of Sensors, Deciders, and Influencers in that configuration, plus one representative Target
- *Number of Links (l)* = the total number of links in the adjacency matrix of the configuration (see Table B-1 for an example matrix)
- *Link to node ratio* = (l/N)
- *Degree distribution* = skewed, since the number of links connected to each node is not uniformly distributed for any of these network configurations, although the degree of skewness does vary)
- *Size, connectivity of largest hubs* = distributed, since the largest hubs (i.e., the Deciders) are not be connected to each other
- *Characteristic path length* = the median of the mean of the lengths of all the shortest paths in the network
- *Clustering coefficient* = 0, since none of the nodes within these network configurations have any direct neighbors that are adjacent to each other (i.e., 3-node cycles)
- *Betweenness* = skewed, since the degree distribution for each configuration is skewed and each path within these configurations is the shortest path, then the betweeness values must be skewed
- *Path horizon* = 1 (since each node within these configurations only needs to interact one adjacent node for consecutive self-synchronization to occur
- *Neutrality rating* = $((l - N + 1)/N)$
- *Coefficient of networked effects* = (λ_{PFE}/N)
- *Susceptibility* = high, given that the dynamic structure of each of these configurations breaks down with the removal of the Deciders.

Table 31. Metric Values for a 7-3-7-1 Network.

Configuration ID #	# of Sensors linked to Decider 1	# of Sensors linked to Decider 2	# of Sensors linked to Decider 3	# of Influencers linked to Decider 1	# of Influencers linked to Decider 2	# of Influencers linked to Decider 3	λ_{PFE} value	Number of Nodes (N)	Number of Links (l)	Link to Node Ratio (l/N)	Degree Distribution	Connectivity of largest hubs	Characteristic Path Length	Clustering Coefficient	Betweeness	Path Horizon	Neutrality Rating ((l - N + 1) / N)	Coefficient of Networked Effects (λ_{PFE}/N)	Susceptibility
0	5	1	1	5	1	1	**2.280**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.127**	high
1	5	1	1	1	5	1	**1.821**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.101**	high
2	5	1	1	4	2	1	**2.190**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.122**	high
3	5	1	1	1	4	2	**1.821**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.101**	high
4	5	1	1	2	1	4	**1.968**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.109**	high
5	5	1	1	3	3	1	**2.088**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.116**	high
6	5	1	1	1	3	3	**1.821**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.101**	high
7	5	1	1	3	2	2	**2.088**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.116**	high
8	5	1	1	2	3	2	**1.968**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.109**	high
9	4	2	1	5	1	1	**2.190**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.122**	high
10	4	2	1	1	5	1	**1.968**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.109**	high
11	4	2	1	1	1	5	**1.821**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.101**	high
12	4	2	1	4	2	1	**2.141**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.119**	high
13	4	2	1	4	1	2	**2.115**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.118**	high
14	4	2	1	2	4	1	**2.031**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.113**	high
15	4	2	1	2	1	4	**1.934**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.107**	high
16	4	2	1	1	4	2	**1.934**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.107**	high
17	4	2	1	1	2	4	**1.861**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.103**	high
18	4	2	1	3	3	1	**2.088**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.116**	high
19	4	2	1	3	1	3	**2.031**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.113**	high
20	4	2	1	1	3	3	**1.899**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.106**	high
21	4	2	1	3	2	2	**2.060**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.114**	high
22	4	2	1	2	3	2	**2.000**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.111**	high
23	4	2	1	2	2	3	**1.968**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.109**	high
24	3	1	3	5	1	1	**2.088**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.116**	high
25	3	1	3	1	5	1	**1.821**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.101**	high
26	3	1	3	4	2	1	**2.031**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.113**	high
27	3	1	3	1	4	2	**1.899**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.106**	high
28	3	1	3	2	1	4	**2.088**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.116**	high
29	3	1	3	3	3	1	**1.968**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.109**	high
30	3	1	3	3	1	3	**2.088**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.116**	high
31	3	1	3	3	2	2	**2.031**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.113**	high
32	3	1	3	2	3	2	**1.968**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.109**	high
33	3	2	2	5	1	1	**2.088**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.116**	high
34	3	2	2	1	5	1	**1.968**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.109**	high
35	3	2	2	4	2	1	**2.060**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.114**	high
36	3	2	2	1	4	2	**1.968**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.109**	high
37	3	2	2	2	1	4	**2.000**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.111**	high
38	3	2	2	3	3	1	**2.031**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.113**	high
39	3	2	2	1	3	3	**1.968**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.109**	high
40	3	2	2	3	2	2	**2.031**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.113**	high
41	3	2	2	2	3	2	**2.000**	18	28	1.556	skewed	distributed	4	0	skewed	1	0.611	**0.111**	high

APPENDIX G

NETLOGO CODE

The agent-based simulation model used for this study was NetLogo (version 4.0.2). This software can be downloaded here: http://ccl.northwestern.edu/netlogo/.

The following code was used to model the meaningful configurations of the 9-5-9-1 networked force. The code used to model the 7-3-7-1 and 8-3-8-1 networked forces differed only in the code necessary to generate the different configurations and in the reduced number of "breeds."

```
breed [ BInfluencer1s BInfluencer1 ]
breed [ BSensor1s BSensor1 ]
breed [ BDecider1s BDecider1 ]
breed [ BInfluencer2s BInfluencer2 ]
breed [ BSensor2s BSensor2 ]
breed [ BDecider2s BDecider2 ]
breed [ BInfluencer3s BInfluencer3 ]
breed [ BSensor3s BSensor3 ]
breed [ BDecider3s BDecider3 ]
breed [ BInfluencer4s BInfluencer4 ]
breed [ BSensor4s BSensor4 ]
breed [ BDecider4s BDecider4 ]
breed [ BInfluencer5s BInfluencer5 ]
breed [ BSensor5s BSensor5 ]
breed [ BDecider5s BDecider5 ]
breed [ RInfluencer1s RInfluencer1 ]
breed [ RSensor1s RSensor1 ]
breed [ RDecider1s RDecider1 ]
breed [ RInfluencer2s RInfluencer2 ]
breed [ RSensor2s RSensor2 ]
breed [ RDecider2s RDecider2 ]
breed [ RInfluencer3s RInfluencer3 ]
breed [ RSensor3s RSensor3 ]
breed [ RDecider3s RDecider3 ]
breed [ RInfluencer4s RInfluencer4 ]
breed [ RSensor4s RSensor4 ]
breed [ RDecider4s RDecider4 ]
breed [ RInfluencer5s RInfluencer5 ]
breed [ RSensor5s RSensor5 ]
breed [ RDecider5s RDecider5 ]
directed-link-breed [detections detection]
directed-link-breed [orders order]
directed-link-breed [LOFs LOF]
globals [BWin RWin Bconfig Rconfig]

BInfluencer1s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
BSensor1s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
```

```
BDecider1s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
BInfluencer2s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
BSensor2s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
BDecider2s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
BInfluencer3s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
BSensor3s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
BDecider3s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
BInfluencer4s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
BSensor4s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
BDecider4s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
BInfluencer5s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
BSensor5s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
BDecider5s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
RInfluencer1s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
RSensor1s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
RDecider1s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
RInfluencer2s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
RSensor2s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
RDecider2s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
RInfluencer3s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
RSensor3s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
RDecider3s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
RInfluencer4s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
RSensor4s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
RDecider4s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
RInfluencer5s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
RSensor5s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
RDecider5s-own [side dead sensedBD1 sensedBD2 sensedBD3 sensedBD4
sensedBD5 sensedRD1 sensedRD2 sensedRD3 sensedRD4 sensedRD5]
```

```
to setup
  clear-all
  random-seed seed
  set-default-shape BInfluencer1s "square"
  set-default-shape BSensor1s "triangle"
  set-default-shape BDecider1s "star"
  set-default-shape BInfluencer2s "square"
  set-default-shape BSensor2s "triangle"
  set-default-shape BDecider2s "star"
  set-default-shape BInfluencer3s "square"
  set-default-shape BSensor3s "triangle"
  set-default-shape BDecider3s "star"
  set-default-shape BInfluencer4s "square"
  set-default-shape BSensor4s "triangle"
  set-default-shape BDecider4s "star"
  set-default-shape BInfluencer5s "square"
  set-default-shape BSensor5s "triangle"
  set-default-shape BDecider5s "star"
  set-default-shape RTargets "circle"
  set-default-shape RInfluencer1s "square"
  set-default-shape RSensor1s "triangle"
  set-default-shape RDecider1s "star"
  set-default-shape RInfluencer2s "square"
  set-default-shape RSensor2s "triangle"
  set-default-shape RDecider2s "star"
  set-default-shape RInfluencer3s "square"
  set-default-shape RSensor3s "triangle"
  set-default-shape RDecider3s "star"
  set-default-shape RInfluencer4s "square"
  set-default-shape RSensor4s "triangle"
  set-default-shape RDecider4s "star"
  set-default-shape RInfluencer5s "square"
  set-default-shape RSensor5s "triangle"
  set-default-shape RDecider5s "star"
  if BID = 0 [set Bconfig [5 1 1 1 1 5 1 1 1 1]]
  if BID = 1 [set Bconfig [5 1 1 1 1 1 5 1 1 1]]
  if BID = 2 [set Bconfig [5 1 1 1 1 4 2 1 1 1]]
  if BID = 3 [set Bconfig [5 1 1 1 1 1 4 2 1 1]]
  if BID = 4 [set Bconfig [5 1 1 1 1 2 1 4 1 1]]
  if BID = 5 [set Bconfig [5 1 1 1 1 3 3 1 1 1]]
  if BID = 6 [set Bconfig [5 1 1 1 1 1 3 3 1 1]]
  if BID = 7 [set Bconfig [5 1 1 1 1 3 2 2 1 1]]
  if BID = 8 [set Bconfig [5 1 1 1 1 1 3 2 2 1]]
  if BID = 9 [set Bconfig [5 1 1 1 1 2 3 2 1 1]]
  if BID = 10 [set Bconfig [5 1 1 1 1 2 2 2 2 1]]
  if BID = 11 [set Bconfig [5 1 1 1 1 1 2 2 2 2]]
  if BID = 12 [set Bconfig [4 2 1 1 1 5 1 1 1 1]]
  if BID = 13 [set Bconfig [4 2 1 1 1 1 5 1 1 1]]
  if BID = 14 [set Bconfig [4 2 1 1 1 1 1 5 1 1]]
  if BID = 15 [set Bconfig [4 2 1 1 1 4 2 1 1 1]]
  if BID = 16 [set Bconfig [4 2 1 1 1 4 1 2 1 1]]
  if BID = 17 [set Bconfig [4 2 1 1 1 2 4 1 1 1]]
  if BID = 18 [set Bconfig [4 2 1 1 1 2 1 4 1 1]]
  if BID = 19 [set Bconfig [4 2 1 1 1 1 4 2 1 1]]
  if BID = 20 [set Bconfig [4 2 1 1 1 1 2 4 1 1]]
  if BID = 21 [set Bconfig [4 2 1 1 1 1 1 4 2 1]]
  if BID = 22 [set Bconfig [4 2 1 1 1 3 3 1 1 1]]
```

```
if BID = 23 [set Bconfig [4 2 1 1 1 3 1 3 1 1]]
if BID = 24 [set Bconfig [4 2 1 1 1 1 3 3 1 1]]
if BID = 25 [set Bconfig [4 2 1 1 1 1 1 3 3 1]]
if BID = 26 [set Bconfig [4 2 1 1 1 3 2 2 1 1]]
if BID = 27 [set Bconfig [4 2 1 1 1 3 1 2 2 1]]
if BID = 28 [set Bconfig [4 2 1 1 1 2 3 2 1 1]]
if BID = 29 [set Bconfig [4 2 1 1 1 1 3 2 2 1]]
if BID = 30 [set Bconfig [4 2 1 1 1 2 2 3 1 1]]
if BID = 31 [set Bconfig [4 2 1 1 1 2 1 3 2 1]]
if BID = 32 [set Bconfig [4 2 1 1 1 1 2 3 2 1]]
if BID = 33 [set Bconfig [4 2 1 1 1 1 1 3 2 2]]
if BID = 34 [set Bconfig [4 2 1 1 1 2 2 2 2 1]]
if BID = 35 [set Bconfig [4 2 1 1 1 2 1 2 2 2]]
if BID = 36 [set Bconfig [4 2 1 1 1 1 2 2 2 2]]
if BID = 37 [set Bconfig [3 3 1 1 1 5 1 1 1 1]]
if BID = 38 [set Bconfig [3 3 1 1 1 1 1 5 1 1]]
if BID = 39 [set Bconfig [3 3 1 1 1 4 2 1 1 1]]
if BID = 40 [set Bconfig [3 3 1 1 1 4 1 2 1 1]]
if BID = 41 [set Bconfig [3 3 1 1 1 2 1 4 1 1]]
if BID = 42 [set Bconfig [3 3 1 1 1 1 1 4 2 1]]
if BID = 43 [set Bconfig [3 3 1 1 1 3 3 1 1 1]]
if BID = 44 [set Bconfig [3 3 1 1 1 3 1 3 1 1]]
if BID = 45 [set Bconfig [3 3 1 1 1 1 1 3 3 1]]
if BID = 46 [set Bconfig [3 3 1 1 1 3 2 2 1 1]]
if BID = 47 [set Bconfig [3 3 1 1 1 3 1 2 2 1]]
if BID = 48 [set Bconfig [3 3 1 1 1 2 2 3 1 1]]
if BID = 49 [set Bconfig [3 3 1 1 1 2 1 3 2 1]]
if BID = 50 [set Bconfig [3 3 1 1 1 1 1 3 2 2]]
if BID = 51 [set Bconfig [3 3 1 1 1 2 2 2 2 1]]
if BID = 52 [set Bconfig [3 3 1 1 1 2 1 2 2 2]]
if BID = 53 [set Bconfig [3 2 2 1 1 5 1 1 1 1]]
if BID = 54 [set Bconfig [3 2 2 1 1 1 5 1 1 1]]
if BID = 55 [set Bconfig [3 2 2 1 1 1 1 1 5 1]]
if BID = 56 [set Bconfig [3 2 2 1 1 4 2 1 1 1]]
if BID = 57 [set Bconfig [3 2 2 1 1 4 1 1 2 1]]
if BID = 58 [set Bconfig [3 2 2 1 1 2 4 1 1 1]]
if BID = 59 [set Bconfig [3 2 2 1 1 2 1 1 4 1]]
if BID = 60 [set Bconfig [3 2 2 1 1 1 4 2 1 1]]
if BID = 61 [set Bconfig [3 2 2 1 1 1 4 1 2 1]]
if BID = 62 [set Bconfig [3 2 2 1 1 1 2 1 4 1]]
if BID = 63 [set Bconfig [3 2 2 1 1 1 1 1 4 2]]
if BID = 64 [set Bconfig [3 2 2 1 1 3 3 1 1 1]]
if BID = 65 [set Bconfig [3 2 2 1 1 3 1 1 3 1]]
if BID = 66 [set Bconfig [3 2 2 1 1 1 3 3 1 1]]
if BID = 67 [set Bconfig [3 2 2 1 1 1 3 1 3 1]]
if BID = 68 [set Bconfig [3 2 2 1 1 1 1 1 3 3]]
if BID = 69 [set Bconfig [3 2 2 1 1 3 2 2 1 1]]
if BID = 70 [set Bconfig [3 2 2 1 1 3 2 1 2 1]]
if BID = 71 [set Bconfig [3 2 2 1 1 3 1 1 2 2]]
if BID = 72 [set Bconfig [3 2 2 1 1 2 3 2 1 1]]
if BID = 73 [set Bconfig [3 2 2 1 1 2 3 1 2 1]]
if BID = 74 [set Bconfig [3 2 2 1 1 2 2 1 3 1]]
if BID = 75 [set Bconfig [3 2 2 1 1 2 1 1 3 2]]
if BID = 76 [set Bconfig [3 2 2 1 1 1 3 2 2 1]]
if BID = 77 [set Bconfig [3 2 2 1 1 1 3 1 2 2]]
if BID = 78 [set Bconfig [3 2 2 1 1 1 2 2 3 1]]
if BID = 79 [set Bconfig [3 2 2 1 1 1 2 1 3 2]]
```

```
if BID = 80 [set Bconfig [3 2 2 1 1 2 2 2 2 1]]
if BID = 81 [set Bconfig [3 2 2 1 1 2 2 1 2 2]]
if BID = 82 [set Bconfig [3 2 2 1 1 1 2 2 2 2]]
if BID = 83 [set Bconfig [2 2 2 2 1 5 1 1 1 1]]
if BID = 84 [set Bconfig [2 2 2 2 1 1 1 1 1 5]]
if BID = 85 [set Bconfig [2 2 2 2 1 4 2 1 1 1]]
if BID = 86 [set Bconfig [2 2 2 2 1 4 1 1 1 2]]
if BID = 87 [set Bconfig [2 2 2 2 1 2 1 1 1 4]]
if BID = 88 [set Bconfig [2 2 2 2 1 3 3 1 1 1]]
if BID = 89 [set Bconfig [2 2 2 2 1 3 1 1 1 3]]
if BID = 90 [set Bconfig [2 2 2 2 1 3 2 2 1 1]]
if BID = 91 [set Bconfig [2 2 2 2 1 3 2 1 1 2]]
if BID = 92 [set Bconfig [2 2 2 2 1 2 2 1 1 3]]
if BID = 93 [set Bconfig [2 2 2 2 1 2 2 2 2 1]]
if BID = 94 [set Bconfig [2 2 2 2 1 2 2 2 1 2]]
set number-BSensor1s item 0 Bconfig
set number-BSensor2s item 1 Bconfig
set number-BSensor3s item 2 Bconfig
set number-BSensor4s item 3 Bconfig
set number-BSensor5s item 4 Bconfig
set number-BInfluencer1s item 5 Bconfig
set number-BInfluencer2s item 6 Bconfig
set number-BInfluencer3s item 7 Bconfig
set number-BInfluencer4s item 8 Bconfig
set number-BInfluencer5s item 9 Bconfig
if RID = 0 [set Rconfig [5 1 1 1 1 5 1 1 1 1]]
if RID = 1 [set Rconfig [5 1 1 1 1 1 5 1 1 1]]
if RID = 2 [set Rconfig [5 1 1 1 1 4 2 1 1 1]]
if RID = 3 [set Rconfig [5 1 1 1 1 1 4 2 1 1]]
if RID = 4 [set Rconfig [5 1 1 1 1 2 1 4 1 1]]
if RID = 5 [set Rconfig [5 1 1 1 1 3 3 1 1 1]]
if RID = 6 [set Rconfig [5 1 1 1 1 1 3 3 1 1]]
if RID = 7 [set Rconfig [5 1 1 1 1 3 2 2 1 1]]
if RID = 8 [set Rconfig [5 1 1 1 1 1 3 2 2 1]]
if RID = 9 [set Rconfig [5 1 1 1 1 2 3 2 1 1]]
if RID = 10 [set Rconfig [5 1 1 1 1 2 2 2 2 1]]
if RID = 11 [set Rconfig [5 1 1 1 1 1 2 2 2 2]]
if RID = 12 [set Rconfig [4 2 1 1 1 5 1 1 1 1]]
if RID = 13 [set Rconfig [4 2 1 1 1 1 5 1 1 1]]
if RID = 14 [set Rconfig [4 2 1 1 1 1 1 5 1 1]]
if RID = 15 [set Rconfig [4 2 1 1 1 4 2 1 1 1]]
if RID = 16 [set Rconfig [4 2 1 1 1 4 1 2 1 1]]
if RID = 17 [set Rconfig [4 2 1 1 1 2 4 1 1 1]]
if RID = 18 [set Rconfig [4 2 1 1 1 2 1 4 1 1]]
if RID = 19 [set Rconfig [4 2 1 1 1 1 4 2 1 1]]
if RID = 20 [set Rconfig [4 2 1 1 1 1 2 4 1 1]]
if RID = 21 [set Rconfig [4 2 1 1 1 1 1 4 2 1]]
if RID = 22 [set Rconfig [4 2 1 1 1 3 3 1 1 1]]
if RID = 23 [set Rconfig [4 2 1 1 1 3 1 3 1 1]]
if RID = 24 [set Rconfig [4 2 1 1 1 1 3 3 1 1]]
if RID = 25 [set Rconfig [4 2 1 1 1 1 1 3 3 1]]
if RID = 26 [set Rconfig [4 2 1 1 1 3 2 2 1 1]]
if RID = 27 [set Rconfig [4 2 1 1 1 3 1 2 2 1]]
if RID = 28 [set Rconfig [4 2 1 1 1 2 3 2 1 1]]
if RID = 29 [set Rconfig [4 2 1 1 1 1 3 2 2 1]]
if RID = 30 [set Rconfig [4 2 1 1 1 2 2 3 1 1]]
if RID = 31 [set Rconfig [4 2 1 1 1 2 1 3 2 1]]
```

```
if RID = 32 [set Rconfig [4 2 1 1 1 1 2 3 2 1]]
if RID = 33 [set Rconfig [4 2 1 1 1 1 1 3 2 2]]
if RID = 34 [set Rconfig [4 2 1 1 1 2 2 2 2 1]]
if RID = 35 [set Rconfig [4 2 1 1 1 2 1 2 2 2]]
if RID = 36 [set Rconfig [4 2 1 1 1 1 2 2 2 2]]
if RID = 37 [set Rconfig [3 3 1 1 1 5 1 1 1 1]]
if RID = 38 [set Rconfig [3 3 1 1 1 1 1 5 1 1]]
if RID = 39 [set Rconfig [3 3 1 1 1 4 2 1 1 1]]
if RID = 40 [set Rconfig [3 3 1 1 1 4 1 2 1 1]]
if RID = 41 [set Rconfig [3 3 1 1 1 2 1 4 1 1]]
if RID = 42 [set Rconfig [3 3 1 1 1 1 1 4 2 1]]
if RID = 43 [set Rconfig [3 3 1 1 1 3 3 1 1 1]]
if RID = 44 [set Rconfig [3 3 1 1 1 3 1 3 1 1]]
if RID = 45 [set Rconfig [3 3 1 1 1 1 1 3 3 1]]
if RID = 46 [set Rconfig [3 3 1 1 1 3 2 2 1 1]]
if RID = 47 [set Rconfig [3 3 1 1 1 3 1 2 2 1]]
if RID = 48 [set Rconfig [3 3 1 1 1 2 2 3 1 1]]
if RID = 49 [set Rconfig [3 3 1 1 1 2 1 3 2 1]]
if RID = 50 [set Rconfig [3 3 1 1 1 1 1 3 2 2]]
if RID = 51 [set Rconfig [3 3 1 1 1 2 2 2 2 1]]
if RID = 52 [set Rconfig [3 3 1 1 1 2 1 2 2 2]]
if RID = 53 [set Rconfig [3 2 2 1 1 5 1 1 1 1]]
if RID = 54 [set Rconfig [3 2 2 1 1 1 5 1 1 1]]
if RID = 55 [set Rconfig [3 2 2 1 1 1 1 1 5 1]]
if RID = 56 [set Rconfig [3 2 2 1 1 4 2 1 1 1]]
if RID = 57 [set Rconfig [3 2 2 1 1 4 1 1 2 1]]
if RID = 58 [set Rconfig [3 2 2 1 1 2 4 1 1 1]]
if RID = 59 [set Rconfig [3 2 2 1 1 2 1 1 4 1]]
if RID = 60 [set Rconfig [3 2 2 1 1 1 4 2 1 1]]
if RID = 61 [set Rconfig [3 2 2 1 1 1 4 1 2 1]]
if RID = 62 [set Rconfig [3 2 2 1 1 1 2 1 4 1]]
if RID = 63 [set Rconfig [3 2 2 1 1 1 1 1 4 2]]
if RID = 64 [set Rconfig [3 2 2 1 1 3 3 1 1 1]]
if RID = 65 [set Rconfig [3 2 2 1 1 3 1 1 3 1]]
if RID = 66 [set Rconfig [3 2 2 1 1 1 3 3 1 1]]
if RID = 67 [set Rconfig [3 2 2 1 1 1 3 1 3 1]]
if RID = 68 [set Rconfig [3 2 2 1 1 1 1 1 3 3]]
if RID = 69 [set Rconfig [3 2 2 1 1 3 2 2 1 1]]
if RID = 70 [set Rconfig [3 2 2 1 1 3 2 1 2 1]]
if RID = 71 [set Rconfig [3 2 2 1 1 3 1 1 2 2]]
if RID = 72 [set Rconfig [3 2 2 1 1 2 3 2 1 1]]
if RID = 73 [set Rconfig [3 2 2 1 1 2 3 1 2 1]]
if RID = 74 [set Rconfig [3 2 2 1 1 2 2 1 3 1]]
if RID = 75 [set Rconfig [3 2 2 1 1 2 1 1 3 2]]
if RID = 76 [set Rconfig [3 2 2 1 1 1 3 2 2 1]]
if RID = 77 [set Rconfig [3 2 2 1 1 1 3 1 2 2]]
if RID = 78 [set Rconfig [3 2 2 1 1 1 2 2 3 1]]
if RID = 79 [set Rconfig [3 2 2 1 1 1 2 1 3 2]]
if RID = 80 [set Rconfig [3 2 2 1 1 2 2 2 2 1]]
if RID = 81 [set Rconfig [3 2 2 1 1 2 2 1 2 2]]
if RID = 82 [set Rconfig [3 2 2 1 1 1 2 2 2 2]]
if RID = 83 [set Rconfig [2 2 2 2 1 5 1 1 1 1]]
if RID = 84 [set Rconfig [2 2 2 2 1 1 1 1 1 5]]
if RID = 85 [set Rconfig [2 2 2 2 1 4 2 1 1 1]]
if RID = 86 [set Rconfig [2 2 2 2 1 4 1 1 1 2]]
if RID = 87 [set Rconfig [2 2 2 2 1 2 1 1 1 4]]
if RID = 88 [set Rconfig [2 2 2 2 1 3 3 1 1 1]]
```

```
if RID = 89 [set Rconfig [2 2 2 2 1 3 1 1 1 3]]
if RID = 90 [set Rconfig [2 2 2 2 1 3 2 2 1 1]]
if RID = 91 [set Rconfig [2 2 2 2 1 3 2 1 1 2]]
if RID = 92 [set Rconfig [2 2 2 2 1 2 2 1 1 3]]
if RID = 93 [set Rconfig [2 2 2 2 1 2 2 2 2 1]]
if RID = 94 [set Rconfig [2 2 2 2 1 2 2 2 1 2]]
set number-RSensor1s item 0 Rconfig
set number-RSensor2s item 1 Rconfig
set number-RSensor3s item 2 Rconfig
set number-RSensor4s item 3 Rconfig
set number-RSensor5s item 4 Rconfig
set number-RInfluencer1s item 5 Rconfig
set number-RInfluencer2s item 6 Rconfig
set number-RInfluencer3s item 7 Rconfig
set number-RInfluencer4s item 8 Rconfig
set number-RInfluencer5s item 9 Rconfig
create-BInfluencer1s number-BInfluencer1s
  [ set size 3
    set color blue
    set side 1   ;; side 1 is all BLUE S and I
    set dead 0
   setxy random-xcor random-ycor ]
create-BSensor1s number-BSensor1s
  [ set size 3
    set color blue
    set side 1   ;; side 1 is all BLUE S and I
    set dead 0
    setxy random-xcor random-ycor ]
create-BDecider1s 1
  [ set size 3
    set color blue
    set side 3   ;; side 3 is all BLUE D
    set dead 0
    setxy random-xcor random-ycor ]
create-BInfluencer2s number-BInfluencer2s
  [ set size 3
    set color blue
    set side 1   ;; side 1 is all BLUE S and I
    set dead 0
    setxy random-xcor random-ycor ]
create-BSensor2s number-BSensor2s
  [ set size 3
    set color blue
    set side 1   ;; side 1 is all BLUE S and I
    set dead 0
    setxy random-xcor random-ycor ]
create-BDecider2s 1
  [ set size 3
    set color blue
    set side 3   ;; side 3 is all BLUE D
    set dead 0
    setxy random-xcor random-ycor ]
create-BInfluencer3s number-BInfluencer3s
  [ set size 3
    set color blue
    set side 1   ;; side 1 is all BLUE S and I
    set dead 0
```

```
    setxy random-xcor random-ycor ]
create-BSensor3s number-BSensor3s
  [ set size 3
    set color blue
    set side 1    ;; side 1 is all BLUE S and I
    set dead 0
    setxy random-xcor random-ycor ]
create-BDecider3s 1
  [ set size 3
    set color blue
    set side 3    ;; side 3 is all BLUE D
    set dead 0
    setxy random-xcor random-ycor ]
create-BInfluencer4s number-BInfluencer4s
  [ set size 3
    set color blue
    set side 1   ;; side 1 is all BLUE S and I
    set dead 0
   setxy random-xcor random-ycor ]
create-BSensor4s number-BSensor4s
  [ set size 3
    set color blue
    set side 1   ;; side 1 is all BLUE S and I
    set dead 0
    setxy random-xcor random-ycor ]
create-BDecider4s 1
  [ set size 3
    set color blue
    set side 3   ;; side 3 is all BLUE D
    set dead 0
    setxy random-xcor random-ycor ]
create-BInfluencer5s number-BInfluencer5s
  [ set size 3
    set color blue
    set side 1   ;; side 1 is all BLUE S and I
    set dead 0
   setxy random-xcor random-ycor ]
create-BSensor5s number-BSensor5s
  [ set size 3
    set color blue
    set side 1   ;; side 1 is all BLUE S and I
    set dead 0
    setxy random-xcor random-ycor ]
create-BDecider5s 1
  [ set size 3
    set color blue
    set side 3   ;; side 3 is all BLUE D
    set dead 0
    setxy random-xcor random-ycor ]
create-RInfluencer1s number-RInfluencer1s
  [ set size 3
    set color red
    set side 2   ;; side 2 is all RED S and I
    set dead 0
    setxy random-xcor random-ycor ]
create-RSensor1s number-RSensor1s
  [ set size 3
```

```
      set color red
      set side 2   ;; side 2 is all RED S and I
      set dead 0
      setxy random-xcor random-ycor ]
  create-RDecider1s 1
    [ set size 3
      set color red
      set side 4   ;; side 4 is all RED D
      set dead 0
      setxy random-xcor random-ycor ]
  create-RInfluencer2s number-RInfluencer2s
    [ set size 3
      set color red
      set side 2   ;; side 2 is all RED S and I
      set dead 0
      setxy random-xcor random-ycor ]
  create-RSensor2s number-RSensor2s
    [ set size 3
      set color red
      set side 2   ;; side 2 is all RED S and I
      set dead 0
      setxy random-xcor random-ycor ]
  create-RDecider2s 1
    [ set size 3
      set color red
      set side 4   ;; side 4 is all RED D
      set dead 0
      setxy random-xcor random-ycor ]
  create-RInfluencer3s number-RInfluencer3s
    [ set size 3
      set color red
      set side 2   ;; side 2 is all RED S and I
      set dead 0
      setxy random-xcor random-ycor ]
  create-RSensor3s number-RSensor3s
    [ set size 3
      set color red
      set side 2   ;; side 2 is all RED S and I
      set dead 0
      setxy random-xcor random-ycor ]
  create-RDecider3s 1
    [ set size 3
      set color red
      set side 4   ;; side 4 is all RED D
      set dead 0
      setxy random-xcor random-ycor ]
  create-RSensor4s number-RSensor4s
    [ set size 3
      set color red
      set side 2   ;; side 2 is all RED S and I
      set dead 0
      setxy random-xcor random-ycor ]
  create-RDecider4s 1
    [ set size 3
      set color red
      set side 4   ;; side 4 is all RED D
      set dead 0
```

```
        setxy random-xcor random-ycor ]
    create-RInfluencer4s number-RInfluencer4s
      [ set size 3
        set color red
        set side 2   ;; side 2 is all RED S and I
        set dead 0
        setxy random-xcor random-ycor ]
    create-RSensor5s number-RSensor5s
      [ set size 3
        set color red
        set side 2   ;; side 2 is all RED S and I
        set dead 0
        setxy random-xcor random-ycor ]
    create-RDecider5s 1
      [ set size 3
        set color red
        set side 4   ;; side 4 is all RED D
        set dead 0
        setxy random-xcor random-ycor ]
    create-RInfluencer5s number-RInfluencer5s
      [ set size 3
        set color red
        set side 2   ;; side 2 is all RED S and I
        set dead 0
        setxy random-xcor random-ycor ]
end

to go
  if count turtles with [(side = 1)] = 0 [   ;; stop when all BLUE S&I
are dead
    set BWin 0
    set RWin 1
    stop ]
  if count turtles with [(side = 2)] = 0 [   ;; stop when all RED S&I
are dead
    set BWin 1
    set RWin 0
    stop ]
  if (ticks = 500) [stop]   ;; stop if there is a standoff (BLUE & RED
cannot shoot due to lack of Sensors)
  establish-links
  sense
  track
  shoot
  kill
  move-Influencer
  move-Influencer
  move-Influencer
  move-Influencer
  move-Influencer
  move-Sensor
  move-Sensor
  move-Sensor
  move-Sensor
  move-Sensor
  reset
  tick
```

```
end

to establish-links   ;; establishes specific S->D and D->I links for
each Decider
  ask BDecider1s [
      ask BSensor1s [create-detection-to myself [set color blue] ]
      ask BInfluencer1s [create-order-from myself [set color blue] ] ]
  ask BDecider2s [
      ask BSensor2s [create-detection-to myself [set color blue] ]
      ask BInfluencer2s [create-order-from myself [set color blue] ] ]
  ask BDecider3s [
      ask BSensor3s [create-detection-to myself [set color blue] ]
      ask BInfluencer3s [create-order-from myself [set color blue] ] ]
  ask BDecider4s [
      ask BSensor4s [create-detection-to myself [set color blue] ]
      ask BInfluencer4s [create-order-from myself [set color blue] ] ]
  ask BDecider5s [
      ask BSensor5s [create-detection-to myself [set color blue] ]
      ask BInfluencer5s [create-order-from myself [set color blue] ] ]
  ask RDecider1s [
      ask RSensor1s [create-detection-to myself [set color red] ]
      ask RInfluencer1s [create-order-from myself [set color red] ] ]
  ask RDecider2s [
      ask RSensor2s [create-detection-to myself [set color red] ]
      ask RInfluencer2s [create-order-from myself [set color red] ] ]
  ask RDecider3s [
      ask RSensor3s [create-detection-to myself [set color red] ]
      ask RInfluencer3s [create-order-from myself [set color red] ] ]
  ask RDecider4s [
      ask RSensor4s [create-detection-to myself [set color red] ]
      ask RInfluencer4s [create-order-from myself [set color red] ] ]
  ask RDecider5s [
      ask RSensor5s [create-detection-to myself [set color red] ]
      ask RInfluencer5s [create-order-from myself [set color red] ] ]
end

to sense   ;; identifies all enemy S & I within s-range of each
Decider's Sensor
  ask BDecider1s [
    ask in-link-neighbors [
      ask RInfluencer1s in-radius s-range [set sensedBD1 1]
      ask RSensor1s in-radius s-range [set sensedBD1 1]
      ask RInfluencer2s in-radius s-range [set sensedBD1 1]
      ask RSensor2s in-radius s-range [set sensedBD1 1]
      ask RInfluencer3s in-radius s-range [set sensedBD1 1]
      ask RSensor3s in-radius s-range [set sensedBD1 1]
      ask RInfluencer4s in-radius s-range [set sensedBD1 1]
      ask RSensor4s in-radius s-range [set sensedBD1 1]
      ask RInfluencer5s in-radius s-range [set sensedBD1 1]
      ask RSensor5s in-radius s-range [set sensedBD1 1] ] ]
  ask BDecider2s [
    ask in-link-neighbors [
      ask RInfluencer1s in-radius s-range [set sensedBD2 1]
      ask RSensor1s in-radius s-range [set sensedBD2 1]
      ask RInfluencer2s in-radius s-range [set sensedBD2 1]
      ask RSensor2s in-radius s-range [set sensedBD2 1]
      ask RInfluencer3s in-radius s-range [set sensedBD2 1]
```

```
      ask RSensor3s in-radius s-range [set sensedBD2 1]
      ask RInfluencer4s in-radius s-range [set sensedBD2 1]
      ask RSensor4s in-radius s-range [set sensedBD2 1]
      ask RInfluencer5s in-radius s-range [set sensedBD2 1]
      ask RSensor5s in-radius s-range [set sensedBD2 1] ] ]
  ask BDecider3s [
    ask in-link-neighbors [
      ask RInfluencer1s in-radius s-range [set sensedBD3 1]
      ask RSensor1s in-radius s-range [set sensedBD3 1]
      ask RInfluencer2s in-radius s-range [set sensedBD3 1]
      ask RSensor2s in-radius s-range [set sensedBD3 1]
      ask RInfluencer3s in-radius s-range [set sensedBD3 1]
      ask RSensor3s in-radius s-range [set sensedBD3 1]
      ask RInfluencer4s in-radius s-range [set sensedBD3 1]
      ask RSensor4s in-radius s-range [set sensedBD3 1]
      ask RInfluencer5s in-radius s-range [set sensedBD3 1]
      ask RSensor5s in-radius s-range [set sensedBD3 1] ] ]
  ask BDecider4s [
    ask in-link-neighbors [
      ask RInfluencer1s in-radius s-range [set sensedBD4 1]
      ask RSensor1s in-radius s-range [set sensedBD4 1]
      ask RInfluencer2s in-radius s-range [set sensedBD4 1]
      ask RSensor2s in-radius s-range [set sensedBD4 1]
      ask RInfluencer3s in-radius s-range [set sensedBD4 1]
      ask RSensor3s in-radius s-range [set sensedBD4 1]
      ask RInfluencer4s in-radius s-range [set sensedBD4 1]
      ask RSensor4s in-radius s-range [set sensedBD4 1]
      ask RInfluencer5s in-radius s-range [set sensedBD4 1]
      ask RSensor5s in-radius s-range [set sensedBD4 1] ] ]
  ask BDecider5s [
    ask in-link-neighbors [
      ask RInfluencer1s in-radius s-range [set sensedBD5 1]
      ask RSensor1s in-radius s-range [set sensedBD5 1]
      ask RInfluencer2s in-radius s-range [set sensedBD5 1]
      ask RSensor2s in-radius s-range [set sensedBD5 1]
      ask RInfluencer3s in-radius s-range [set sensedBD5 1]
      ask RSensor3s in-radius s-range [set sensedBD5 1]
      ask RInfluencer4s in-radius s-range [set sensedBD5 1]
      ask RSensor4s in-radius s-range [set sensedBD5 1]
      ask RInfluencer5s in-radius s-range [set sensedBD5 1]
      ask RSensor5s in-radius s-range [set sensedBD5 1] ] ]
  ask RDecider1s [
    ask in-link-neighbors [
      ask BInfluencer1s in-radius s-range [set sensedRD1 1]
      ask BSensor1s in-radius s-range [set sensedRD1 1]
      ask BInfluencer2s in-radius s-range [set sensedRD1 1]
      ask BSensor2s in-radius s-range [set sensedRD1 1]
      ask BInfluencer3s in-radius s-range [set sensedRD1 1]
      ask BSensor3s in-radius s-range [set sensedRD1 1]
      ask BInfluencer4s in-radius s-range [set sensedRD1 1]
      ask BSensor4s in-radius s-range [set sensedRD1 1]
      ask BInfluencer5s in-radius s-range [set sensedRD1 1]
      ask BSensor5s in-radius s-range [set sensedRD1 1] ] ]
  ask RDecider2s [
    ask in-link-neighbors [
      ask BInfluencer1s in-radius s-range [set sensedRD2 1]
      ask BSensor1s in-radius s-range [set sensedRD2 1]
```

```
      ask BInfluencer2s in-radius s-range [set sensedRD2 1]
      ask BSensor2s in-radius s-range [set sensedRD2 1]
      ask BInfluencer3s in-radius s-range [set sensedRD2 1]
      ask BSensor3s in-radius s-range [set sensedRD2 1]
      ask BInfluencer4s in-radius s-range [set sensedRD2 1]
      ask BSensor4s in-radius s-range [set sensedRD2 1]
      ask BInfluencer5s in-radius s-range [set sensedRD2 1]
      ask BSensor5s in-radius s-range [set sensedRD2 1] ] ]
  ask RDecider3s [
    ask in-link-neighbors [
      ask BInfluencer1s in-radius s-range [set sensedRD3 1]
      ask BSensor1s in-radius s-range [set sensedRD3 1]
      ask BInfluencer2s in-radius s-range [set sensedRD3 1]
      ask BSensor2s in-radius s-range [set sensedRD3 1]
      ask BInfluencer3s in-radius s-range [set sensedRD3 1]
      ask BSensor3s in-radius s-range [set sensedRD3 1]
      ask BInfluencer4s in-radius s-range [set sensedRD3 1]
      ask BSensor4s in-radius s-range [set sensedRD3 1]
      ask BInfluencer5s in-radius s-range [set sensedRD3 1]
      ask BSensor5s in-radius s-range [set sensedRD3 1] ] ]
  ask RDecider4s [
    ask in-link-neighbors [
      ask BInfluencer1s in-radius s-range [set sensedRD4 1]
      ask BSensor1s in-radius s-range [set sensedRD4 1]
      ask BInfluencer2s in-radius s-range [set sensedRD4 1]
      ask BSensor2s in-radius s-range [set sensedRD4 1]
      ask BInfluencer3s in-radius s-range [set sensedRD4 1]
      ask BSensor3s in-radius s-range [set sensedRD4 1]
      ask BInfluencer4s in-radius s-range [set sensedRD4 1]
      ask BSensor4s in-radius s-range [set sensedRD4 1]
      ask BInfluencer5s in-radius s-range [set sensedRD4 1]
      ask BSensor5s in-radius s-range [set sensedRD4 1] ] ]
  ask RDecider5s [
    ask in-link-neighbors [
      ask BInfluencer1s in-radius s-range [set sensedRD5 1]
      ask BSensor1s in-radius s-range [set sensedRD5 1]
      ask BInfluencer2s in-radius s-range [set sensedRD5 1]
      ask BSensor2s in-radius s-range [set sensedRD5 1]
      ask BInfluencer3s in-radius s-range [set sensedRD5 1]
      ask BSensor3s in-radius s-range [set sensedRD5 1]
      ask BInfluencer4s in-radius s-range [set sensedRD5 1]
      ask BSensor4s in-radius s-range [set sensedRD5 1]
      ask BInfluencer5s in-radius s-range [set sensedRD5 1]
      ask BSensor5s in-radius s-range [set sensedRD5 1] ] ]
end

to track   ;; links all Influencers to enemy S &I within i-range
  ask BDecider1s [
    ask out-link-neighbors [
      ask RInfluencer1s in-radius i-range [create-LOF-from myself]
      ask RSensor1s in-radius i-range [create-LOF-from myself]
      ask RInfluencer2s in-radius i-range [create-LOF-from myself]
      ask RSensor2s in-radius i-range [create-LOF-from myself]
      ask RInfluencer3s in-radius i-range [create-LOF-from myself]
      ask RSensor3s in-radius i-range [create-LOF-from myself]
      ask RInfluencer4s in-radius i-range [create-LOF-from myself]
      ask RSensor4s in-radius i-range [create-LOF-from myself]
```

```
      ask RInfluencer4s in-radius i-range [create-LOF-from myself]
      ask RSensor4s in-radius i-range [create-LOF-from myself] ] ]
  ask BDecider2s [
    ask out-link-neighbors [
      ask RInfluencer1s in-radius i-range [create-LOF-from myself]
      ask RSensor1s in-radius i-range [create-LOF-from myself]
      ask RInfluencer2s in-radius i-range [create-LOF-from myself]
      ask RSensor2s in-radius i-range [create-LOF-from myself]
      ask RInfluencer3s in-radius i-range [create-LOF-from myself]
      ask RSensor3s in-radius i-range [create-LOF-from myself]
      ask RInfluencer4s in-radius i-range [create-LOF-from myself]
      ask RSensor4s in-radius i-range [create-LOF-from myself]
      ask RInfluencer5s in-radius i-range [create-LOF-from myself]
      ask RSensor5s in-radius i-range [create-LOF-from myself] ] ]
  ask BDecider3s [
    ask out-link-neighbors [
      ask RInfluencer1s in-radius i-range [create-LOF-from myself]
      ask RSensor1s in-radius i-range [create-LOF-from myself]
      ask RInfluencer2s in-radius i-range [create-LOF-from myself]
      ask RSensor2s in-radius i-range [create-LOF-from myself]
      ask RInfluencer3s in-radius i-range [create-LOF-from myself]
      ask RSensor3s in-radius i-range [create-LOF-from myself]
      ask RInfluencer4s in-radius i-range [create-LOF-from myself]
      ask RSensor4s in-radius i-range [create-LOF-from myself]
      ask RInfluencer5s in-radius i-range [create-LOF-from myself]
      ask RSensor5s in-radius i-range [create-LOF-from myself] ] ]
  ask BDecider4s [
    ask out-link-neighbors [
      ask RInfluencer1s in-radius i-range [create-LOF-from myself]
      ask RSensor1s in-radius i-range [create-LOF-from myself]
      ask RInfluencer2s in-radius i-range [create-LOF-from myself]
      ask RSensor2s in-radius i-range [create-LOF-from myself]
      ask RInfluencer3s in-radius i-range [create-LOF-from myself]
      ask RSensor3s in-radius i-range [create-LOF-from myself]
      ask RInfluencer4s in-radius i-range [create-LOF-from myself]
      ask RSensor4s in-radius i-range [create-LOF-from myself]
      ask RInfluencer5s in-radius i-range [create-LOF-from myself]
      ask RSensor5s in-radius i-range [create-LOF-from myself] ] ]
  ask BDecider5s [
    ask out-link-neighbors [
      ask RInfluencer1s in-radius i-range [create-LOF-from myself]
      ask RSensor1s in-radius i-range [create-LOF-from myself]
      ask RInfluencer2s in-radius i-range [create-LOF-from myself]
      ask RSensor2s in-radius i-range [create-LOF-from myself]
      ask RInfluencer3s in-radius i-range [create-LOF-from myself]
      ask RSensor3s in-radius i-range [create-LOF-from myself]
      ask RInfluencer4s in-radius i-range [create-LOF-from myself]
      ask RSensor4s in-radius i-range [create-LOF-from myself]
      ask RInfluencer5s in-radius i-range [create-LOF-from myself]
      ask RSensor5s in-radius i-range [create-LOF-from myself] ] ]
  ask RDecider1s [
    ask out-link-neighbors [
      ask BInfluencer1s in-radius i-range [create-LOF-from myself]
      ask BSensor1s in-radius i-range [create-LOF-from myself]
      ask BInfluencer2s in-radius i-range [create-LOF-from myself]
      ask BSensor2s in-radius i-range [create-LOF-from myself]
      ask BInfluencer3s in-radius i-range [create-LOF-from myself]
```

```
      ask BSensor3s in-radius i-range [create-LOF-from myself]
      ask BInfluencer4s in-radius i-range [create-LOF-from myself]
      ask BSensor4s in-radius i-range [create-LOF-from myself]
      ask BInfluencer5s in-radius i-range [create-LOF-from myself]
      ask BSensor5s in-radius i-range [create-LOF-from myself] ] ]
  ask RDecider2s [
    ask out-link-neighbors [
      ask BInfluencer1s in-radius i-range [create-LOF-from myself]
      ask BSensor1s in-radius i-range [create-LOF-from myself]
      ask BInfluencer2s in-radius i-range [create-LOF-from myself]
      ask BSensor2s in-radius i-range [create-LOF-from myself]
      ask BInfluencer3s in-radius i-range [create-LOF-from myself]
      ask BSensor3s in-radius i-range [create-LOF-from myself]
      ask BInfluencer4s in-radius i-range [create-LOF-from myself]
      ask BSensor4s in-radius i-range [create-LOF-from myself]
      ask BInfluencer5s in-radius i-range [create-LOF-from myself]
      ask BSensor5s in-radius i-range [create-LOF-from myself] ] ]
  ask RDecider3s [
    ask out-link-neighbors [
      ask BInfluencer1s in-radius i-range [create-LOF-from myself]
      ask BSensor1s in-radius i-range [create-LOF-from myself]
      ask BInfluencer2s in-radius i-range [create-LOF-from myself]
      ask BSensor2s in-radius i-range [create-LOF-from myself]
      ask BInfluencer3s in-radius i-range [create-LOF-from myself]
      ask BSensor3s in-radius i-range [create-LOF-from myself]
      ask BInfluencer4s in-radius i-range [create-LOF-from myself]
      ask BSensor4s in-radius i-range [create-LOF-from myself]
      ask BInfluencer5s in-radius i-range [create-LOF-from myself]
      ask BSensor5s in-radius i-range [create-LOF-from myself] ] ]
  ask RDecider4s [
    ask out-link-neighbors [
      ask BInfluencer1s in-radius i-range [create-LOF-from myself]
      ask BSensor1s in-radius i-range [create-LOF-from myself]
      ask BInfluencer2s in-radius i-range [create-LOF-from myself]
      ask BSensor2s in-radius i-range [create-LOF-from myself]
      ask BInfluencer3s in-radius i-range [create-LOF-from myself]
      ask BSensor3s in-radius i-range [create-LOF-from myself]
      ask BInfluencer4s in-radius i-range [create-LOF-from myself]
      ask BSensor4s in-radius i-range [create-LOF-from myself]
      ask BInfluencer5s in-radius i-range [create-LOF-from myself]
      ask BSensor5s in-radius i-range [create-LOF-from myself] ] ]
  ask RDecider5s [
    ask out-link-neighbors [
      ask BInfluencer1s in-radius i-range [create-LOF-from myself]
      ask BSensor1s in-radius i-range [create-LOF-from myself]
      ask BInfluencer2s in-radius i-range [create-LOF-from myself]
      ask BSensor2s in-radius i-range [create-LOF-from myself]
      ask BInfluencer3s in-radius i-range [create-LOF-from myself]
      ask BSensor3s in-radius i-range [create-LOF-from myself]
      ask BInfluencer4s in-radius i-range [create-LOF-from myself]
      ask BSensor4s in-radius i-range [create-LOF-from myself]
      ask BInfluencer5s in-radius i-range [create-LOF-from myself]
      ask BSensor5s in-radius i-range [create-LOF-from myself] ] ]
end

to shoot   ;;  identifies the nearest enemy S & I that has been
"sensed" and "tracked" by a friendly S &I linked to the same Decider
```

```
  ask BDecider1s [
    ask out-link-neighbors [
      ask out-link-neighbors [
        let $targets-sensed turtles with [(sensedBD1 = 1) and (side =
2)]
          if any? $targets-sensed [
            ask min-one-of $targets-sensed [distance myself] [set dead
1] ] ] ] ]
  ask BDecider2s [
    ask out-link-neighbors [
      ask out-link-neighbors [
        let $targets-sensed turtles with [(sensedBD2 = 1) and (side =
2)]
          if any? $targets-sensed [
            ask min-one-of $targets-sensed [distance myself] [set dead
1] ] ] ] ]
  ask BDecider3s [
    ask out-link-neighbors [
      ask out-link-neighbors [
        let $targets-sensed turtles with [(sensedBD3 = 1) and (side =
2)]
          if any? $targets-sensed [
            ask min-one-of $targets-sensed [distance myself] [set dead
1] ] ] ] ]
  ask BDecider4s [
    ask out-link-neighbors [
      ask out-link-neighbors [
        let $targets-sensed turtles with [(sensedBD4 = 1) and (side =
2)]
          if any? $targets-sensed [
            ask min-one-of $targets-sensed [distance myself] [set dead
1] ] ] ] ]
  ask BDecider5s [
    ask out-link-neighbors [
      ask out-link-neighbors [
        let $targets-sensed turtles with [(sensedBD5 = 1) and (side =
2)]
          if any? $targets-sensed [
            ask min-one-of $targets-sensed [distance myself] [set dead
1] ] ] ] ]
  ask RDecider1s [
    ask out-link-neighbors [
      ask out-link-neighbors [
        let $targets-sensed turtles with [(sensedRD1 = 1) and (side =
1)]
          if any? $targets-sensed [
            ask min-one-of $targets-sensed [distance myself] [set dead
1] ] ] ] ]
  ask RDecider2s [
    ask out-link-neighbors [
      ask out-link-neighbors [
        let $targets-sensed turtles with [(sensedRD2 = 1) and (side =
1)]
          if any? $targets-sensed [
            ask min-one-of $targets-sensed [distance myself] [set dead
1] ] ] ] ]
  ask RDecider3s [
```

```
    ask out-link-neighbors [
      ask out-link-neighbors [
        let $targets-sensed turtles with [(sensedRD3 = 1) and (side =
1)]
          if any? $targets-sensed [
            ask min-one-of $targets-sensed [distance myself] [set dead
1] ] ] ] ]
  ask RDecider4s [
    ask out-link-neighbors [
      ask out-link-neighbors [
        let $targets-sensed turtles with [(sensedRD4 = 1) and (side =
1)]
          if any? $targets-sensed [
            ask min-one-of $targets-sensed [distance myself] [set dead
1] ] ] ] ]
  ask RDecider5s [
    ask out-link-neighbors [
      ask out-link-neighbors [
        let $targets-sensed turtles with [(sensedRD5 = 1) and (side =
1)]
          if any? $targets-sensed [
            ask min-one-of $targets-sensed [distance myself] [set dead
1] ] ] ] ]
end

to kill   ;; kills all turtles that have been "sensed", "tracked" and
"shot"
  ask turtles with [(dead = 1)] [die]
end

to move-Influencer   ;;  moves all I towards the nearest enemy S or I
within a related Sensors's s-range
  ask BDecider1s [
    ask out-link-neighbors [
      let $targets-sensed turtles with [(sensedBD1 = 1) and (side = 2)]
        if any? $targets-sensed [
          set heading towards min-one-of $targets-sensed [distance
myself] forward 1 ] ] ]
  ask RDecider1s [
    ask out-link-neighbors [
      let $targets-sensed turtles with [(sensedRD1 = 1) and (side = 1)]
        if any? $targets-sensed [
          set heading towards min-one-of $targets-sensed [distance
myself] forward 1 ] ] ]
  ask BDecider2s [
    ask out-link-neighbors [
      let $targets-sensed turtles with [(sensedBD2 = 1) and (side = 2)]
        if any? $targets-sensed [
          set heading towards min-one-of $targets-sensed [distance
myself] forward 1 ] ] ]
  ask RDecider2s [
    ask out-link-neighbors [
      let $targets-sensed turtles with [(sensedRD2 = 1) and (side = 1)]
        if any? $targets-sensed [
          set heading towards min-one-of $targets-sensed [distance
myself] forward 1 ] ] ]
  ask BDecider3s [
```

```
    ask out-link-neighbors [
      let $targets-sensed turtles with [(sensedBD3 = 1) and (side = 2)]
        if any? $targets-sensed [
          set heading towards min-one-of $targets-sensed [distance
myself] forward 1 ] ] ]
  ask RDecider3s [
    ask out-link-neighbors [
      let $targets-sensed turtles with [(sensedRD3 = 1) and (side = 1)]
        if any? $targets-sensed [
          set heading towards min-one-of $targets-sensed [distance
myself] forward 1 ] ] ]
  ask BDecider4s [
    ask out-link-neighbors [
      let $targets-sensed turtles with [(sensedBD4 = 1) and (side = 2)]
        if any? $targets-sensed [
          set heading towards min-one-of $targets-sensed [distance
myself] forward 1 ] ] ]
  ask RDecider4s [
    ask out-link-neighbors [
      let $targets-sensed turtles with [(sensedRD4 = 1) and (side = 1)]
        if any? $targets-sensed [
          set heading towards min-one-of $targets-sensed [distance
myself] forward 1 ] ] ]
  ask BDecider5s [
    ask out-link-neighbors [
      let $targets-sensed turtles with [(sensedBD5 = 1) and (side = 2)]
        if any? $targets-sensed [
          set heading towards min-one-of $targets-sensed [distance
myself] forward 1 ] ] ]
  ask RDecider5s [
    ask out-link-neighbors [
      let $targets-sensed turtles with [(sensedRD5 = 1) and (side = 1)]
        if any? $targets-sensed [
          set heading towards min-one-of $targets-sensed [distance
myself] forward 1 ] ] ]
end

to move-Sensor   ;; moves all S without a "sensed" enemy S or I towards
the nearest "unsensed" (by that Decider's Sensors!) enemy S or I
  ask BDecider1s [
    ask in-link-neighbors [
      let $targets-sensed turtles with [(sensedBD1 = 1) and (side = 2)]
        if not any? $targets-sensed [
          let $targets-unsensed turtles with [(sensedBD1 = 0) and (side
= 2)]
            if any? $targets-unsensed [
              let $nearest-unsensed min-one-of $targets-unsensed
[distance myself]
              set heading towards $nearest-unsensed
              forward 1 ] ] ] ]
  ask RDecider1s [
    ask in-link-neighbors [
      let $targets-sensed turtles with [(sensedRD1 = 1) and (side = 1)]
        if not any? $targets-sensed [
          let $targets-unsensed turtles with [(sensedRD1 = 0) and (side
= 1)]
            if any? $targets-unsensed [
```

```
              let $nearest-unsensed min-one-of $targets-unsensed
[distance myself]
              set heading towards $nearest-unsensed
              forward 1 ] ] ] ]
  ask BDecider2s [
    ask in-link-neighbors [
      let $targets-sensed turtles with [(sensedBD2 = 1) and (side = 2)]
        if not any? $targets-sensed [
          let $targets-unsensed turtles with [(sensedBD2 = 0) and (side
= 2)]
            if any? $targets-unsensed [
              let $nearest-unsensed min-one-of $targets-unsensed
[distance myself]
              set heading towards $nearest-unsensed
              forward 1 ] ] ] ]
  ask RDecider2s [
    ask in-link-neighbors [
      let $targets-sensed turtles with [(sensedRD2 = 1) and (side = 1)]
        if not any? $targets-sensed [
          let $targets-unsensed turtles with [(sensedRD2 = 0) and (side
= 1)]
            if any? $targets-unsensed [
              let $nearest-unsensed min-one-of $targets-unsensed
[distance myself]
              set heading towards $nearest-unsensed
              forward 1 ] ] ] ]
  ask BDecider3s [
    ask in-link-neighbors [
      let $targets-sensed turtles with [(sensedBD3 = 1) and (side = 2)]
        if not any? $targets-sensed [
          let $targets-unsensed turtles with [(sensedBD3 = 0) and (side
= 2)]
            if any? $targets-unsensed [
              let $nearest-unsensed min-one-of $targets-unsensed
[distance myself]
              set heading towards $nearest-unsensed
              forward 1 ] ] ] ]
  ask RDecider3s [
    ask in-link-neighbors [
      let $targets-sensed turtles with [(sensedRD3 = 1) and (side = 1)]
        if not any? $targets-sensed [
          let $targets-unsensed turtles with [(sensedRD3 = 0) and (side
= 1)]
            if any? $targets-unsensed [
              let $nearest-unsensed min-one-of $targets-unsensed
[distance myself]
              set heading towards $nearest-unsensed
              forward 1 ] ] ] ]
  ask BDecider4s [
    ask in-link-neighbors [
      let $targets-sensed turtles with [(sensedBD4 = 1) and (side = 2)]
        if not any? $targets-sensed [
          let $targets-unsensed turtles with [(sensedBD4 = 0) and (side
= 2)]
            if any? $targets-unsensed [
              let $nearest-unsensed min-one-of $targets-unsensed
[distance myself]
```

```
              set heading towards $nearest-unsensed
              forward 1 ] ] ] ]
  ask RDecider4s [
    ask in-link-neighbors [
      let $targets-sensed turtles with [(sensedRD4 = 1) and (side = 1)]
        if not any? $targets-sensed [
          let $targets-unsensed turtles with [(sensedRD4 = 0) and (side
= 1)]
            if any? $targets-unsensed [
              let $nearest-unsensed min-one-of $targets-unsensed
[distance myself]
              set heading towards $nearest-unsensed
              forward 1 ] ] ] ]
  ask BDecider5s [
    ask in-link-neighbors [
      let $targets-sensed turtles with [(sensedBD5 = 1) and (side = 2)]
        if not any? $targets-sensed [
          let $targets-unsensed turtles with [(sensedBD5 = 0) and (side
= 2)]
            if any? $targets-unsensed [
              let $nearest-unsensed min-one-of $targets-unsensed
[distance myself]
              set heading towards $nearest-unsensed
              forward 1 ] ] ] ]
  ask RDecider5s [
    ask in-link-neighbors [
      let $targets-sensed turtles with [(sensedRD5 = 1) and (side = 1)]
        if not any? $targets-sensed [
          let $targets-unsensed turtles with [(sensedRD5 = 0) and (side
= 1)]
            if any? $targets-unsensed [
              let $nearest-unsensed min-one-of $targets-unsensed
[distance myself]
              set heading towards $nearest-unsensed
              forward 1 ] ] ] ]
end

to reset   ;; clears all "sensed" and "tracked" targets, and all links
  ask turtles [
    set sensedBD1 0
    set sensedBD2 0
    set sensedBD3 0
    set sensedBD4 0
    set sensedBD5 0
    set sensedRD1 0
    set sensedRD2 0
    set sensedRD3 0
    set sensedRD4 0
    set sensedRD5 0]
  ask links [die]
end
```

VITA

Lieutenant Colonel Sean Deller was born in Baldwin, New York on December 24, 1965. He graduated from the United States Military Academy in 1988 with a B.S. in Mathematics and was commissioned as a Second Lieutenant in the Aviation branch of the United States Army. He received a M.E. in Engineering Management from Old Dominion University in 1999, and he conducted this research while a student of the Department of Engineering Management and Systems Engineering, Old Dominion University, Kaufman Hall, Room 241, Norfolk, VA , 23529.

During his 21 years of active service, he has served in Aviation units in Korea, Egypt and Fort Stewart, Georgia. He became an operations research analyst in 1997 and has served at the United States Army Training and Doctrine Command, and at the Joint Center for Operational Analyses and the Joint Experimentation Directorate of the United States Joint Forces Command. He is now serving as the Capabilities Assessment Branch Chief at the Army Capabilities Integration Center at Fort Monroe, Virginia. His operational assignments include deployments in support of Operation Enduring Freedom in Afghanistan, Operation Iraqi Freedom in Iraq, and the NATO Kosovo Force in Kosovo.

His military education includes the Aviation Basic and Advanced Courses, Airborne School, Flight School, and the Command and General Staff Officer Course.

His military awards include the Bronze Star.